Efficient facial expression recognition using decision trees and neural networks

Gogić, Ivan

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivan Gogić

EFFICIENT FACIAL EXPRESSION RECOGNITION USING DECISION TREES AND NEURAL NETWORKS

DOCTORAL THESIS

Zagreb, 2021

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivan Gogić

EFFICIENT FACIAL EXPRESSION RECOGNITION USING DECISION TREES AND NEURAL NETWORKS

DOCTORAL THESIS

Supervisors: Professor Igor S. Pandžić, Ph.D.
Assistant Professor Jörgen Ahlberg, Ph.D.

Zagreb, 2021

Sveučilište u Zagrebu
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivan Gogić

UČINKOVITO RASPOZNAVANJE IZRAZA LICA PRIMJENOM STABALA ODLUČIVANJA I NEURONSKIH MREŽA

DOKTORSKI RAD

Mentori: Prof. dr. sc. Igor S. Pandžić
Doc. dr. sc. Jörgen Ahlberg

Zagreb, 2021.

The doctoral thesis was completed at the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Telecommunications and at the company Visage Technologies.

Supervisor: Professor Igor S. Pandžić, Ph.D.
Department of Telecommunications,
Faculty of Electrical Engineering and Computing,
University of Zagreb

Supervisor: Assistant Professor Jörgen Ahlberg, Ph.D.
Computer Vision Laboratory,
Department of Electrical Engineering,
Linköping University

The thesis has: 106 pages

Thesis number: _____

O mentorima

Igor S. Pandžić redoviti je profesor na Zavodu za telekomunikacije Fakulteta elektrotehnike i računarstva na Sveučilištu u Zagrebu. Voditelj je istraživačkog laboratorija Human-Oriented Technologies Laboratory (HOTLab). Predaje preddiplomske i diplomske predmete u području virtualnih okruženja i komunikacije. Glavno područje njegovog istraživačkog interesa je računalna grafika uz dodatak računalnog vida sa posebnim naglaskom na analizu i animaciju lica te primjenu tih tehnologija. Igor je također radio na umreženim virtualnim okruženjima, računalno generiranoj filmskoj produkciji i paralelnom računarstvu. Objavio je pet knjiga i oko 100 članaka u spomenutim područjima.

Jörgen Ahlberg je diplomirao 1996. godine, a doktorsku disertaciju obranio 2002. godine na Sveučilištu u Linköpingu. Nakon toga radio je devet godina kao znanstvenik i voditelj istraživanja u Švedskoj agenciji za obranu (FOI). Trenutno je izvanredni profesor na Sveučilištu u Linköpingu te vodi projekte razvoja i istraživanja u Visage Technologies, Termisk Systemteknik i Glana Sensors, tvrtkama specijaliziranim za optički, termalni i hiperspektralni računalni vid. Njegovi istraživački interesi uključuju praćenje i analizu slika ljudskog lica iz generalnog područja analize slika i računalnog vida te automatsku detekciju, prepoznavanje i praćenje u termalnim i hiperspektralnim sustavima. Objavio je više od 60 znanstvenih radova, od kojih su četiri dobila nagrade, i šest patenata.

About the Supervisors

Igor S. Pandžić is a Professor at the Department of Telecommunications, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia. He leads the Human-Oriented Technologies Laboratory (HOTLab). He teaches undergraduate and postgraduate courses in the fields of virtual environments and communications. His main research interests are in the field of computer graphics and, more recently, computer vision, with particular interest in face analysis and animation and strong focus on applications of these technologies. Igor also worked on networked collaborative virtual environments, computer generated film production and parallel computing. He published five books and around 100 papers on these topics.

Jörgen Ahlberg received his M.Sc degree in Computer Science and Engineering in 1996 and his Ph.D in Electrical Engineering in 2002, both from Linköping University, Sweden. He then held positions as scientist and research leader at FOI, the Swedish Defence Research Agency for nine years. He is currently an Adjunct Senior Lecturer at Linköping University and runs R&D projects at Visage Technologies, Termisk Systemteknik, and Glana Sensors, companies

specializing in visual, thermal and hyperspectral computer vision respectively. Research interests, in the general area of image analysis and vision, includes tracking and analysis of facial images as well as automatic detection, recognition, and tracking in thermal and hyperspectral systems. He has published more than 60 scientific papers, of which four are award-winning, and been granted 6 patents.

Acknowledgements

The author thanks his supervisors, Igor S. Pandžić, PhD and Jörgen Ahlberg, PhD, for providing an encouraging work environment. Their support was essential for the realization of the work presented in this thesis. The author also thanks his colleagues from Visage Technologies and his Algo team for their support with a special thanks to:

- Martina and Bruno for their help with the training framework implementation;
- Krešimir for numerous insightful discussions;

Last but not least, the author thanks his parents for their support, his wife Sara and daughter Elena for putting up with his frequent absent-mindedness, and his newborn son Nikola for pushing him over the finish line.

Abstract

This thesis investigates a facial expression recognition system that estimates the emotional state of subjects from facial images. Such systems demand accurate and fast algorithms that can run in real-time on platforms with limited computational resources. The proposed algorithms bridge the gap between precise but slow methods and fast but less precise methods, combining decision trees and neural networks. The gentle boost decision trees are trained to extract highly discriminative feature vectors for each facial expression around distinct facial landmark points. These sparse binary features are concatenated to jointly optimize facial expression predictions with a shallow neural network architecture. The joint optimization improves the recognition rates of difficult expressions such as fear and sadness. Since the algorithm depends on accurate landmark locations, a novel face alignment method is introduced using gradient boost decision trees and neural networks organized in a cascaded regression framework. The cascade is initialized by a lightweight convolutional neural network to increase robustness while preserving high efficiency. The thesis begins with an introduction to the problem and the motivation for solving it, followed by an explanation of the theoretical background and a systematic overview of related, previous work. Next, novel algorithms for face alignment and facial expression recognition are described and evaluated on relevant public data sets. The results demonstrate high efficiency and competitive accuracy compared to the state-of-the-art methods suitable for power-efficient applications. The final chapter provides concluding remarks of the thesis.

Keywords: decision trees, neural networks, convolutional neural networks, facial expression recognition, face alignment

Prošireni sažetak

Učinkovito raspoznavanje izraza lica primjenom stabala odlučivanja i neuronskih mreža

Doktorski rad podijeljen je u šest poglavlja. Prvo poglavlje donosi uvod u temu raspoznavanja izraza lica i povezanu temu detekcije ključnih točaka lica, motivaciju za rješavanje tih problema te opisuje glavne primjene u industriji. U istom poglavlju navedeni su i najvažniji doprinosi doktorskog rada. U drugom poglavlju kratko su opisane metode strojnog učenja korištene u ostatku doktorskog rada. Treće poglavlje pruža sistematski pregled dosadašnjeg rada na područjima detekcije ključnih točaka i raspoznavanja izraza lica. Četvrto i peto poglavlje opisuju nove metode za detekciju karakterističnih točaka lica i raspoznavanje izraza lica te dobivene rezultate. Šesto i posljednje poglavlje donosi glavne zaključke doktorskog rada. Više detalja slijedi u nastavku.

Prvo poglavlje – Uvod

Opažanje vidom jedan je od najbitnijih načina kojim ljudi doživljavaju svijet oko sebe. U globalnim naporima postizanja umjetne inteligencije, važan je korak omogućiti računalima opažanje i doživljavanje okoline kroz istraživanje u području računalnog vida. Tradicionalne tehnike obrade slike korištene su za uobičajene probleme detekcije i prepoznavanja predmeta u slici. Međutim, nedavno su se počeli primjenjivati algoritmi strojnog učenja u zadacima računalnog vida sa porastom dostupnosti podataka i naporima ručnog označavanja. Strojno učenje proširilo se poljem računalnog vida iskorištavajući ubrzani napredak dubokog učenja i konvolucijskih neuronskih mreža.

Jedan od najzanimljivijih objekata ljudima i računalima za analizu je ljudsko lice. Mnogo izazovnih zadataka računalnog vida vezanih uz lice predstavljeno je tokom godina istraživanja uključujući detekciju i praćenje lica, raspoznavanje identiteta, dobi, spola i osjećaja. Ovaj rad usredotočen je na raspoznavanje osjećaja ili, točnije, izraza lica iz slike. Ekman i Friesen precizno su utvrdili kategorije izraza lica neovisnih o kulturi i podrijetlu. Iako postoje drugačije definicije osjećaja, u ovom radu koristimo šest osnovnih koji se najčešće koriste u zajednici

računalnog vida: ljutnja, gađenje, strah, sreća, tuga i iznenađenost.

Važan korak pri točnom i učinkovitom raspoznavanju izraza lica je detekcija ključnih točaka lica (omeđuju oči, nos, usta, obrve, bradu i rub lica) iz slike uz poznati položaj i veličinu lica. Takve točke opisuju oblik lica te su predstavljene vektorom 2D koordinata. Algoritmi strojnog učenja koriste se za rješavanje ovog problema koji se proučava i u ovom radu jer je sastavni dio sustava za raspoznavanje izraza lica.

Raspoznavanje izraza lica jedan je od osnovnih izazova polja afektivnog računarstva sa mogućim primjenama u sljedećim područjima (među ostalima): industrija zabave i računalnih igara, istraživanje u oglašavanju, usluge maloprodaje, psihologija, robotika. Uvelike se iščekuje skora promjena načina suradnje s računalima uz korištenje afektivnih aplikacija no još uvijek je izazovno izgraditi takve sustave. Raspoznavanje izraza lica ključni je dio takvih sustava zbog velikog udjela neverbalnog izražavanja u ljudskoj komunikaciji. Upravo zbog toga, istraživačka zajednica uložila je velike napore u razvoj metoda za raspoznavanje izraza iz slika i videa.

Međutim, problem je vrlo izazovan prvenstveno zbog kompleksnosti koju uvodi kulturološka i individualna raznolikost uz različite uvjete snimanja (položaj lica, osvjetljenje, prekrivenost lica). Uz širenje mobilnih i drugih pametnih uređaja niske energetske potrošnje u sklopu interneta stvari, učinkovitost algoritama računalnog vida postaje dodatna važna mjera kvalitete. Dakle, postoji potreba za učinkovitim i točnim algoritmima.

Nedavno je pokazano da značajke izlučene u okolini ključnih točaka lica značajno doprinose točnosti klasifikacije. Izvlačenje značajki iz lokalnih regija bitnih područja lica pomaže smanjenju velikog bazena mogućih značajki te usredotočavanju algoritma na diskriminativna područja lica. Zato je bitno točno odrediti položaj ključnih točaka lica koje se često koriste kao osnova i za druge algoritme. Određivanje oblika lica nužno je za raspoznavanje identiteta osobe jer se koristi kao korak predobrade kako bi se lice registriralo i podesilo uklanjajući time rotacije u ravnini slike te pružajući dosljedne isječke lica za daljnju obradu. Položaji ključnih točaka lica koriste se i kao osnovna komponenta za procjenu 3D poze glave. Samostalno se primjenjuje u raznim aplikacijama kao što su video konferencijski pozivi, računalne igre, animacije i proširena stvarnost.

U ovom radu istražuje se učinkovita kombinacija stabala odlučivanja i neuronskih mreža sa primjenom na usko povezane zadatke detekcije ključnih točaka lica i raspoznavanja izraza lica. Ukratko, glavni doprinosi rada su sljedeći:

- Metoda za klasifikaciju ili regresiju na temelju slike koja sjedinjuje neuronske mreže sa skupovima stabala odlučivanja za izlučivanje značajki koji su sposobni kroz učenje prilagoditi se problemu.
- Primjena predložene metode na problem raspoznavanja izraza lica uz poboljšanje u vremenu izvođenja za red veličine u odnosu na dosadašnje istraživanje.
- Metoda kaskadne regresije za izlučivanje značajki lica koja kombinira konvolucijske neu-

ronske mreže za globalnu inicijalizaciju značajki sa lokalnim binarnim značajkama za fino podešavanje, a koja metoda postiže visoku točnost uz malo vrijeme izvođenja.

Drugo poglavlje – Teorijske osnove

Metode strojnog učenja postale su osnovni alat za rješavanje problema računalnog vida. Glavni cilj procesa strojnog učenja je proizvesti algoritam sposoban riješiti određeni zadatak bez tradicionalnog programiranja. Za to su potrebni podaci kako bi se izveo matematički model kroz optimizaciju kriterijske funkcije koja predstavlja cilj zadatka. Metode strojnog učenja mogu se podijeliti u sljedeće kategorije: nadzirano, nenadzirano i podržano učenje. Glavna razlika je u formulaciji cilja učenja. Nenadzirano učenje ne koristi ručno označene podatke, a cilj je otkriti grupe podataka sličnih uzoraka. Metode podržanog učenja trenirane su sa povratnom informacijom više razine koristeći nagrade i kazne slično ljudskom procesu učenja. Oblikovanje kriterijske funkcije u tom slučaju je fleksibilnije u usporedbi sa uobičajenim nadziranim učenjem te primjenjivo na nediferencijabilne parametre učenja.

Većina algoritama strojnog učenja ipak pripada tradicionalnom nadziranom obliku koji se dijeli na klasifikacijske i regresijske probleme. Obje grupe trebaju precizne ručne oznake željenog rezultata koje u slučaju klasifikacijskih algoritama pripadaju diskretnoj distribuciji. S druge strane, regresijski izlazi predstavljaju kontinuiranu numeričku vrijednost određenog raspona. Također se razlikuju po drugačije oblikovanoj kriterijskoj funkciji koja se optimizira pomoću promjenjivih parametara u procesu učenja.

Stabla odlučivanja uobičajeno se koriste kao alat za podršku i vizualizaciju u analizi odluka. Međutim, mnogo zanimljivija upotreba iz perspektive ovog rada je u obliku modela za predviđanje u statistici, rudarenju podataka i strojnom učenju. Najčešće korištena inačica su binarna stabla odlučivanja gdje svaki čvor roditelj ima dva čvora djeteta uz binarni uvjet grananja. Takvi modeli uče se u nadziranom obliku te se primjenjuju na regresijske i klasifikacijske probleme. Binarni uvjeti ili testovi u svakom čvoru vrše hijerarhijsku podjelu podataka za učenje dok čvorovi listovi sadrže rezultat predviđanja. Parametri testova u čvorovima mijenjaju se kako bi minimizirali kriterijsku funkciju koja predstavlja čistoću dobivene podjele podataka.

U radu se opisuje proces gradnje stabla odlučivanja korištenjem umjetnog regresijskog problema u ilustrativne svrhe. Umjetni podaci dobiveni su sinusoidnom funkcijom sa dodatkom nasumičnog šuma u signalu. Istražuju se različite dubine stabala te prikazuje problem pretjerane prilagodbe stabla podacima sa povećanjem broja čvorova te posljedično modeliranje šuma u signalu.

Umjesto gradnje kompleksnijih stabala odlučivanja sklonih pretjeranoj prilagodbi koristi se kombinacija više manjih i slabijih stabala ograničene dubine u obliku ansambla. Dvije uobičajene grupe metoda formiranja ansambla su *bagging* i *boosting*. *Bagging* je tehnika izgradnje stabla korištenjem nasumično odabranog podskupa podataka za trening. *Boosting* je slijedna

tehnika izgradnje stabala uzimajući u obzir greške prethodnih stabala u slijedu.

Prednosti stabala odlučivanja kao alata strojnog učenja su jednostavnost i lakoća upotrebe. Za razliku od ostalih metoda, stabla mogu kombinirati različite tipove značajki bez normalizacije. Predviđanja se lako tumače slijeđenjem lanca odluka. Njihova nelinearnost pruža snažnu fleksibilnost prilikom modeliranja različitih distribucija podataka. Glavni nedostatak je njihova sklonost pretjeranoj prilagodbi koja zahtjeva kompleksnije tehnike i namještanje hiperparametara.

Dodatni nedostatak metoda ansambla, poglavito *random forest* metode, je neovisno i pohlepno učenje svakog stabla zasebno koja rješavaju svoj lokalni podskup problema. Dobiveni rezultati su uprosječeni bez potencijalne sinergije komplementarnih informacija. U doktorskom radu opisan je prijedlog rješenja spomenutog problema objavljen na CVPR-u 2015. godine (Ren et al.). Glavna ideja je reformulirati predviđanja ansambla u linearni oblik pomoću binarnog vektora indikacije te matrica težina. Koristeći takav oblik, predviđanja ansambla mogu se optimizirati linearnim metodama na globalan način uzimajući u obzir sva stabla u ansamblu.

U kontekstu računalnog vida, stabla odlučivanja mogu koristiti učinkovite i jednostavne binarne testove usporedbe intenziteta piksela. U kombinaciji sa opisanom reformulacijom, ansambli stabala odlučivanja mogu se promatrati kao specijalizirani proces izvlačenja značajki iz slike. Promatrajući globalnu optimizaciju iz takve perspektive, moć predviđanja modela može se unaprijediti korištenjem neuronskih mreža umjesto linearne optimizacije.

Umjetne neuronske mreže su računalni modeli djelomično nadahnuti biološkim živčanim sustavom sa prvim implementacijama još u 1940-im godinama. Praktična primjenjivost takvih modela se značajno unaprijedila u zadnjem desetljeću sa napretkom računalne moći i paralelnog izvršavanja korištenjem grafičkih kartica. Umjetne neuronske mreže i njihove varijante su trenutno dominantni alati u područjima umjetne inteligencije i računalnog vida.

Osnovne komponente neuronskih mreža su neuroni i njihove veze. Sa stanovišta teorije grafova, to su jednostavno čvorovi i bridovi koji tvore usmjeren graf sa težinama. Svaki čvor prima više ulaza, a daje jedan izlaz. Ulazi mogu biti značajke izvučene iz podataka ili izlazi drugih čvorova. Otežani zbroj svih ulaza u čvor tvori aktivacijsku vrijednost koja prolazi kroz nelinearnu aktivacijsku funkciju kako bi se dobio krajnji izlaz. Čvorovi su obično grupirani u sljedne slojeve, a zadnji sloj proizvodi predviđanje modela neuronske mreže.

Središnja komponenta nadziranog učenja neuronskih mreža je algoritam povratnog postupka (engl. *backpropagation*). Koristi se za učinkovito računanje gradijenta kriterijske funkcije s obzirom na težine mreže korištenjem pravila lanca (engl. *chain rule*). Nadalje, to omogućuje korištenje optimizacijskih metoda temeljenih na gradijentu kako bi se minimizirala kriterijska funkcija te model naučio predviđati određene izlaze ručno označenih podataka.

Uz napretke u optimizacijskim metodama, mnogo truda uloženo je i u istraživanje različitih arhitektura umjetnih neuronskih mreža. Najvažniji arhitekturalni napredak za računalni vid je

izum kovolucijskih neuronskih mreža po uzoru na vizualni korteks u mozgu. Naivna primjena neuronskih mreža na slikama dovodi do eksplozije broja parametara modela čak i na najmanjim rezolucijama zbog visoke dimenzionalnosti slika kao ulaza. Konvolucijski slojevi koriste filtre sa dijeljenim težinama umjesto potpuno spojenog sloja koji se pomiču i primjenjuju preko cijele slike korak po korak. Ušteda u broju parametara omogućuje duže slijedove takvih slojeva te otvara put dubokim arhitekturama i novom području dubokog učenja (engl. *deep learning*). Dodatna prednost takvih arhitektura je prilagodba filtera podacima čime se generiraju značajke niske (rubovi i mrlje) i visoke (relevantne abstrakcije) razine kroz duboki slijed slojeva. Takve značajke pokazale su se superiornima tradicionalnim ručno osmišljenim značajkama. U kombinaciji sa paralelnim izvođenjem na grafičkim karticama, duboko učenje prevladalo je područjem računalnog vida u zadnjem desetljeću.

Treće poglavlje – Povezani radovi

U ovom poglavlju predstavljeni su relevantni prijašnji radovi na temu detekcije ključnih točaka lica i raspoznavanja izraza lica. Obzirom da su položaji ključnih točaka preduvjet mnogih problema vezanih uz lice, istraživačka zajednica posvetila je više pozornosti tom problemu te zahtjeva temeljitiji pregled od raspoznavanja izraza lica.

Diskriminativne metode temeljene na regresiji procjenjuju položaje ključnih točaka neposredno iz slike lica. Ova grupa metoda pokazala je nadmoćnu točnost, brzinu i robusnost u odnosu na ranije generativne metode. Koristi se uobičajena regresijska formulacija problema gdje su ciljne vrijednosti vektor razlika između početnih i ručno označenih položaja točaka, a ulaz značajke izlučene iz slike. Početni položaji su obično usrednjeni položaji podataka za učenje normalizirani obzirom na okvir lica.

Metode ograničene regresije uče zasebno predvidjeti položaje pojedinih ključnih točaka iz značajki slika uz korektivni korak koji osigurava globalni oblik lica svih točaka. Ova grupa metoda je najraniji pokušaj robusne detekcije ključnih točaka lica. Uskoro je pokazano da lokalni izgled točke, iako vrlo važan, nije dovoljan za točnu lokalizaciju. Jednako su važne informacije susjednih točaka te globalna konfiguracija oblika lica kako bi se riješile ekstremne varijacije izgleda lica. Vrlo je teško ručno konstruirati ograničenja koja će dozvoliti sve moguće varijacije, a ujedno pružiti potrebnu robusnost.

Kaskadna regresija prometnula se u vodeći pristup za detekciju ključnih točaka lica zbog svoje učinkovitosti, robusnosti i točnosti. U sklopu ovog pristupa, nekoliko regresora koristi se slijedno od početne procjene položaja. Svaki regresor uči procijeniti pomak te osvježava trenutne položaje točaka. Značajke se izlučuju u svakom stupnju kaskade koristeći okolinu slike trenutnih položaja ključnih točaka. Nekoliko je značajnih poboljšanja uvedeno ovim pristupom. Globalna informacija oblika lica se više ne konstruira ručno nego se implicitno uvodi kroz učenje iz podataka poboljšavajući time sposobnost generalizacije. Nadalje, lokalizacija

ključnih točaka uči se zajednički, a ne zasebno, koristeći pritom lokalne značajke i informacije susjednih točaka. Na kraju, kompleksnost problema razlomljena je na niz jednostavnijih problema kroz kaskadnu arhitekturu. Raniji stupnjevi kaskade usredotočeni su na grubu lokalizaciju dok kasniji stupnjevi vrše fino pozicioniranje uz fokus na lokalne detalje.

Nedostatak kaskadne regresije je korištenje lokalnih značajki koje nemaju dovoljno konteksta za primjere sa inicijalnim položajima daleko od ručno označenih. Takvi primjeri su česti kod skupova podataka skupljenih u neograničenim uvjetima. Ti nedostaci rješavaju se inicijalizacijom pomoću početnog globalnog stupnja kaskade koji koristi cjelovitu sliku lica i konvolucijske neuronske mreže većeg kapaciteta. Drugi pristup je korištenje paralelnih kaskada modela treniranih na specifičnim podskupovima problema te njihovo udruživanje za konačno predviđanje. Mana takvog pristupa je povećana potrošnja računalnih resursa.

Kao i u ostalim problemima računalnog vida, duboko učenje koristi se i za detekciju ključnih točaka lica. Primjena, međutim, nije jednostavna zbog nedostatka velikih skupova podataka koji su često potrebni za duboko učenje. U mnogim radovima koriste se različita arhitekturalna rješenja ili načini objedinjavanja podataka sa različitim oznakama kako bi se zaobišao ovaj problem. Također, koristi se i kaskadna arhitektura sa modelima dubokog učenja uz prijenos znanja između stupnjeva. Najnoviji pristup je korištenje arhitekture pješčanog sata uz toplinsku mapu kao izlaz čime se dobije i vjerojatnost točnosti predviđanja.

Nearhitekturalni smjerovi razvoja ovog područja uključuju direktno korištenje 3D položaja točaka, istovremeno učenje više povezanih zadataka te modeliranje prekrivenosti lica objektom. Obzirom da je lice 3D objekt, korištenjem poluručno označenih 3D točaka lica omogućuje se detekcija na puno većem rasponu rotacija lica te povećava iskoristivost u raznim položajima kamere u odnosu na lice. Mnogi zadaci vezani uz lice (detekcija lica, određivanje poze lica, raspoznavanje izraza itd.) koriste slično znanje izvučeno iz slike te postoji uzajamna korist zajednički ih učiti sa istim modelom. Mnogo radova različitih razina uspješnosti provedeno je i na tu temu. Naočale, kape, šalovi i drugi objekti često prekrivaju lice u stvarnom svijetu te se očekuje od algoritama za detekciju ključnih točaka određena razina otpornosti na takve pojave. S napretkom algoritama, ovakvo svojstvo otpornosti dolazi u centar pažnje istraživanja.

Kao što je već spomenuto u uvodu, za raspoznavanje izraza lica koristi se šest osnovnih emocija pogodnih za automatsko raspoznavanje. Tradicionalno se postupak dijeli na tri dijela: detekcija lica, izlučivanje značajki i klasifikacija. U većini slučajeva položaj i veličina lica se uzimaju kao unaprijed poznate informacije te se ne ulazi u detalje detekcije lica. Najveći naglasak stavlja se na odabir i izlučivanje značajki jer se često smatraju kritičnim dijelom sustava dok se uobičajene metode strojnog učenja koriste za klasifikacijski korak. Korištene značajke temelje se na izgledu ili geometriji ključnih točaka lica. Značajke temeljene na izgledu izlučuju se iz slike lica te predstavljaju diskriminativni uzorak teksture dok su geometrijske značajke vrlo osjetljive na individualni oblik lica te manje dosljedne kod različitih osoba.

Raniji pristupi koriste ručno izrađene značajke izlučene iz cijele slike ili pravilne mreže dijelova slike lica, ali postignuti su i napretci u određivanju zajedničkih i posebno bitnih dijelova lica za svaki izraz. U takvim radovima pokazana je važnost detekcije ključnih točaka lica kako bi se pronašle značajne regije lica za izlučivanje značajki. S druge strane, koristi se i sjedinjavanje različitih algoritama izlučivanja značajki i geometrijskih značajki. U zadnje vrijeme, međutim, koristi se duboko učenje i konvolucijske neuronske mreže. Kako bi se spriječio problem pretjerane prilagodbe podacima za trening, tipičan za duboko učenje i nedostatak podataka, koriste se različiti pristupi: nadopuna umjetnim podacima, spajanje različitih skupova podataka te prijenos znanja.

Četvrto poglavlje – Globalno inicijalizirana detekcija ključnih točaka lica primjenom lokalnih binarnih značajki

Cilj metode predložene u ovom poglavlju je učinkovita procjena 2D položaja ključnih točaka na slici lica uz poznati položaj i veličinu lica. Arhitektura kaskadne regresije pokazala je povoljnu ravnotežu između efikasnosti i točnosti uz dvije ključne stavke: duboka globalna inicijalizacija i lokalne značajke trenutnih točaka.

Duboko učenje pogodno je za grubu lokalizaciju ključnih točaka iz nekoliko razloga. Prvo, konvolucijske neuronske mreže koriste globalne značajke koje uzimaju u obzir cijelo lice te njegovo okruženje. Time je olakšano predviđanje orijentacije lica i glave. Drugo, konvolucijske neuronske mreže posjeduju visoku sposobnost prilagodljivosti i apsorbiranja ekstremnih odstupanja u izgledu zbog različitih pozadina i položaja glave. Najbitnija stavka komponente globalne inicijalizacije je odabir arhitekture neuronske mreže. Razvoj arhitektura najčešće je motiviran generalnim problemom detekcije i klasifikacije objekata što rezultira kompleksnim dubokim arhitekturama zbog brojnosti i različitosti kategorija objekata uz milijune primjera. Za globalnu inicijalizaciju predložena je jednostavna arhitektura nadahnuta računalno nezahtjevnom metodom za detekciju objekata YOLO9000 sa milijun parametara koja je prilagođena problemu detekcije ključnih točaka lica.

Postepena prilagodba točaka lica pospješuje točnost i robusnost na što ukazuje brojnost metoda kaskadne regresije. Međutim, lokalne značajke vezane za trenutno predviđene položaje točaka su također bitne jer pružaju algoritmu način usredotočenja nužan za fino prilagođavanje. Ograničenje prostora mogućih značajki korištenjem lokalnih područja oko grubo pozicioniranih ključnih točaka omogućuje učinkovito izlučivanje bitnih značajki. Predložena metoda to ostvaruje korištenjem značajki razlike intenziteta piksela i ansambla stabala odlučivanja. Ansambli stabala reformuliraju se u linearni oblik pomoću kojeg se mogu izlučiti lokalne binarne značajke. Korištenjem neuronske mreže oblika uskog grla vrši se globalna optimizacija predviđenih pomaka ključnih točaka. Time se postižu značajne uštede memorije i vremena izvođenja u

odnosu na izvorni linearni oblik.

Predložena metoda testirana je na 300-W skupu podataka koji se smatra mjerilom za usporedbu točnosti detekcije ključnih točaka lica. Rezultati pokazuju iznimnu robusnost predložene metode na zahtjevne poze, osvjetljenja i prekrivenost lica sa stopom neuspjeha od 1.45%. Predložena metoda postiže konkurentne rezultate u usporedbi sa drugim metodama. Uzimajući u obzir i vrijeme izvođenja, postignut je izvrstan omjer točnosti i efikasnosti sa vremenom izvođenja od 3 ms.

Peto poglavlje – Raspoznavanje izraza lica primjenom lokalnih binarnih značajki i plitkih neuronskih mreža

Cilj metode predložene u ovom poglavlju je raspoznavanje šest osnovnih izraza iz slike lica. Metoda koristi značajke izgleda zbog veće robusnosti na različite oblike lica u usporedbi sa geometrijskim značajkama uz dvije ključne komponente: učenje lokalnih značajki i zajednička klasifikacija izraza.

Kao i kod mnogih drugih problema računalnog vida, duboko učenje prevladava nad tradicionalnim metodama koje koriste ručno dizajnirane značajke za raspoznavanje izraza lica. Proces je, međutim, sporiji zbog nedostatka podataka. Predloženi pristup ublažava taj problem korištenjem stabala odlučivanja optimiziranih za izlučivanje relevantnih značajki na malom broju primjera. Dodatna prednost je korištenje ključnih točaka lica kako bi se izlučile značajke iz bitnih dijelova lica što dodatno poboljšava efikasnost na malim skupovima podataka. Stabla odlučivanja treniraju se za svaki izraz zasebno metodom "jedan naprama svih" u okolini svake ključne točke pomoću razlike u intenzitetima piksela. Već opisanom linearnom reformulacijom dobivaju se lokalne binarne značajke koje se spajaju u jedan globalni binarni vektor. Pomoću njega vrši se zajednička klasifikacija izraza lica.

Prednost zajedničke klasifikacije krije se u činjenici da izrazi lica nisu međusobno neovisni. Neki od njih se mogu kombinirati dijeleći pritom određene pokrete mišića lica, dok su neki izrazi međusobno isključivi što naznačuje kompleksnije skrivene odnose. Predložena metoda koristi neuronsku mrežu uključujući njene nelinearnosti kako bi modelirala te odnose i poboljšala točnost klasifikacije. Uz korištenje značajki izlučenih pomoću stabala odlučivanja, plitka arhitektura neuronske mreže sa jednim skrivenim slojem pokazala se dovoljnom za visoku razinu točnosti. Eksperimentalno je pokazano kako izrazima tuge i straha najviše pogoduje zajednička klasifikacija izraza lica.

Opisana metoda testirana je na četiri često korištena skupa podataka: CK+, MMI, JAFFE i SFEW 2.0. Većina eksperimenata izvršena je korištenjem deseterostruke unakrsne provjere valjanosti zbog malog broja primjera u skupovima podataka. Nadalje, svi eksperimenti su podijeljeni u scenarije ovisne i neovisne o osobi u skupu podataka pri čemu je neovisni scenarij

kompleksniji jer se testira na osobama koje nisu prisutne u fazi treninga. Također, testirana je sposobnost generalizacije metode treniranjem na jednom skupu, a testiranjem na drugom skupu podataka. Metoda postiže najvišu točnost na najčešće korištenom skupu CK+, visoku razinu generalizacije te snažnu robusnost na podacima SFEW 2.0 skupljenim u nekontroliranim uvjetima. Uz visoku točnost, metoda postiže nisko vrijeme izvođenja od 1 ms bez paralelizacije što predstavlja ubrzanje za red veličine u odnosu na prethodne radove. Točnost i brzina izvođenja čine metodu idealnom za uređaje sa ograničenim računalnim sredstvima te prikladnom zamjenom algoritmima dubokog učenja u slučajevima ograničenog broja podataka i računalnih sredstava.

Šesto poglavlje – Zaključci

Zadnje poglavlje donosi pregled postignutih rezultata i doprinosa doktorskog rada. Predloženi algoritmi čine učinkovit sustav raspoznavanja izraza lica prikladan uređajima niske potrošnje energije i ograničenih računalnih sredstava. Također, predložena sprega stabala odlučivanja i neuronskih mreža pokazuje visoku razinu rasuđivanja na manjim skupovima podataka te je iskoristiva alternativa dubokom učenju.

Ključne riječi: stabla odlučivanja, neuronske mreže, usporedbe intenziteta piksela, poravnavanje lica, detekcija ključnih točaka, raspoznavanje izraza lica

Contents

Chapter 1

Introduction

Visual perception is arguably one of the most important ways in which we humans interpret the world around us. In a global effort to achieve artificial intelligence, an important step is to enable the computers to perceive and interpret their environment, intensively researched in the computer vision field. Traditionally, image processing techniques have been used to solve the common problems of detecting and recognizing objects in the image. Recently, however, machine learning algorithms have been used to recognize and detect objects in the image with the rise of data availability and manual annotation efforts. It has dominated the computer vision field on almost every problem riding on the very recent advances in deep learning and convolutional neural networks (CNN).

One of the most attractive objects to analyze for both humans and computers is the human face. Many challenging computer vision problems regarding the human face have been introduced over the years including face detection and tracking, identity recognition, age, gender, and emotion estimation (an example is shown in Figure 1.1). This thesis focuses on emotion estimation or, to be more precise, facial expression recognition from images.

In order to automatically recognize emotions and their related expressions, an investigation on how to define those terms needed to be done first. In [1], Ekman and Friesen discovered six basic or prototypic emotions (anger, disgust, fear, happiness, sadness, and surprise) whose facial expressions are culturally and racially invariant and are, therefore, great candidates for automatic systems which need clear categories. Although other representations have been used as well, in this thesis, we focus on the six basic expressions classification approach as it is currently the most widely used categorization in the computer vision community.

An important stepping stone to achieving accurate and efficient emotion and expression recognition [2, 3] is face alignment. It is the process of determining the face shape, i.e., the location of characteristic facial features or landmarks (points that delineate eyes, nose, mouth, eyebrows, chin, and face contour) given a face image. A vast majority of face alignment methods assume that the face bounding box is known both at training and testing phases. The face

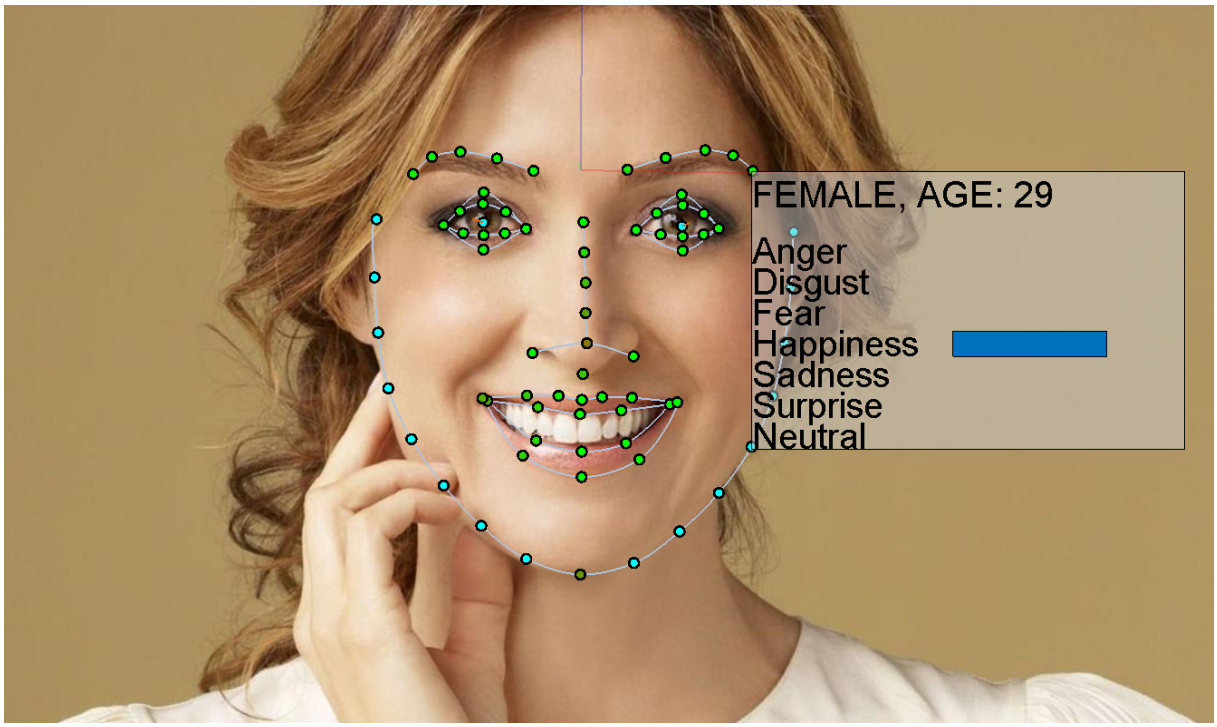


Figure 1.1: An example of a face analysis system using an image of a face.

bounding box is usually obtained through face detection algorithms (see [4]) or from manual annotations ("ground truth"). The configuration of facial landmarks, also known as face shape, is represented as a vector of 2D landmark coordinates. Various machine learning algorithms can be employed to estimate the face shape. If we denote it with $S = (x_1, y_1, \ldots, x_L, y_L)$ where L represents the number of landmarks, the goal of face alignment, given a face image, is to find a shape S closest to the ground truth shape S^*. More formally, the goal is to minimize:

$$||S - S^*|| \tag{1.1}$$

where $||\cdot||$ is a suitable vector norm. The alignment error in (1.1) is used as a performance measure that drives the training process. Face alignment is also studied in this thesis since it is an integral part of a facial expression recognition system.

1.1 Motivation

Facial expression recognition is one of the basic challenges in the affective computing field with potential applications in entertainment, marketing research, retail, psychology, and other domains. It has been widely expected that affect-sensitive applications may change the way we interact with computers [5] yet it remains a challenge to build such systems. Facial expression recognition is an especially important part of these systems since a large segment of human interactions are conveyed non-verbally [6]. Therefore, the research community recently

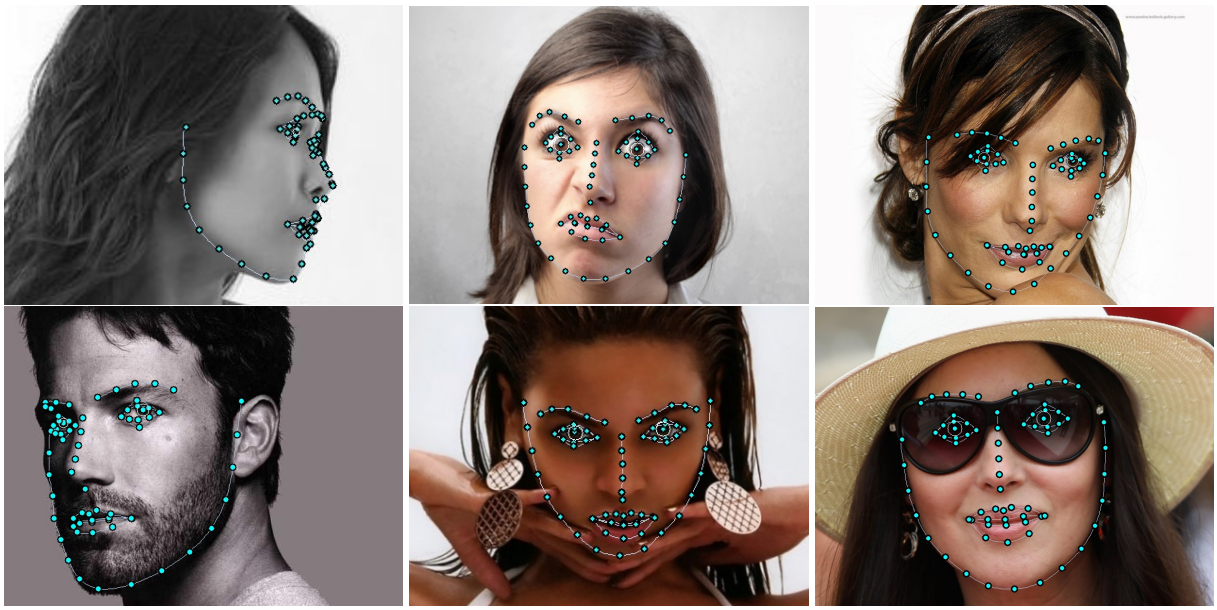


Figure 1.2: Examples of face alignment in large variations of head pose, occlusion level, expression and illumination.

invested extensive efforts to produce methods that can robustly extract expressions from images or videos.

However, numerous challenges still lay ahead primarily due to the complex nature of the problem at hand in the form of large cultural and personal variations in addition to variations in imaging conditions (face pose, lighting, occlusions, etc.) as shown in Figure 1.2. With the proliferation of mobile and other low-powered smart devices within the internet of things (IoT) framework, the computing efficiency of computer vision algorithms becomes an increasingly important parameter along with standard accuracy measurements. Therefore, an accurate yet highly efficient algorithm is needed.

Traditional FER systems consist of three steps: face detection, feature extraction, and classification. However, with recent advances in deep learning algorithms, end-to-end convolutional neural networks have become prevalent in many computer vision fields. Their distinct, competitive advantage is the joint optimization of both feature extraction (through convolution filters' weights) and classification (through fully connected layers' weights). The largest obstacle, however, is the need for extremely large data sets to prevent over-fitting of deep networks. FER data sets are hard to collect due to the ethical issues of eliciting negative emotions (fear, anger, sadness) and the difficulty to act and annotate accompanying expressions. Therefore, an algorithm that can learn to extract custom task-specific features from a limited number of samples per expression would be beneficial.

As recently demonstrated, appearance features extracted around facial landmarks significantly contribute to the classification accuracy [7, 8]. Given the positions of important facial regions, extracting features from local patches can help reduce the extremely large pool of pos-

sible features and focus the algorithm on discriminative regions of the face. It is, therefore, important to accurately locate facial landmarks also used as a base for other algorithms. For face recognition, face alignment is necessary as a preprocessing step to register and align facial images eliminating in-plane rotations and providing consistent facial crops for further processing [9, 10, 11]. Detected facial landmark points are the key component to correctly estimate the parameters of a 3D Morphable Model (3DMM) [12] providing 3D head pose and facial action units as a result [13].

Additionally, face alignment has application areas in many different industries, including human-machine interaction, video conferencing, gaming, animation, and augmented reality. The applications range from fun, augmented-reality gimmicks such as face masking or virtual make-up to life-saving technology in the automotive industry like driver distraction and drowsiness detection (examples are shown in Figure 1.3). For all these reasons, it rightfully received attention from the computer vision research community.

1.2 About this thesis

This thesis investigates an efficient combination of decision trees and neural networks applied to the tightly connected problems of face alignment and facial expression recognition. The proposed algorithm uses simple pixel difference features coupled with ensembles of decision trees [14] to train and extract highly discriminative shape-indexed local binary features (LBF). The extracted features represent task-relevant patterns used together with a neural network for final classification and regression.

When applied to facial expression recognition, the algorithm enables the neural network to model non-linear interactions between expressions improving the recognition accuracy. Additionally, highly discriminative features are extracted from salient regions of the face made possible by the detected landmark locations. For face alignment, the algorithm is organized in a cascade, allowing gradual regression and pattern re-sampling at each stage. By introducing a bottleneck-shaped neural network architecture, the execution time and memory consumption of the original method [15] are further improved. Furthermore, the cascade is initialized with a lightweight CNN architecture using global features, ensuring robustness on challenging examples.

The presented novel methods form an efficient facial expression recognition system suitable for platforms with limited computing power, as will be experimentally demonstrated. The main contributions of this thesis are as follows:

- A method for image-based classification or regression combining neural networks with ensembles of decision trees for task-specific feature extraction.
- Application of the proposed method to the facial expression recognition problem giving

Figure 1.3: Examples of face tracking applications in different industries: driver monitoring system, face masking, virtual make-up, and marketing research.

an order of magnitude improvement in execution time compared to previous research.

- A cascaded regression method for face alignment combining CNNs for global feature initialization with local binary features for fine alignment, achieving high accuracy and low execution time.

1.3 Organization of the thesis

The rest of the thesis is organized in the following way. Chapter 2 briefly introduces the concept of machine learning and basic methods used throughout the thesis. In chapter 3, related work is introduced for both facial expression recognition and face alignment. The described algorithms are systematically analyzed to provide the reasoning behind the proposed methods in chapters 4 and 5. The novel methods using decision trees and neural networks for face alignment and facial expression recognition problems are described in chapters 4 and 5, respectively. The methods are experimentally verified and compared with the state-of-the-art on relevant benchmark data sets. The final chapter 6 concludes the thesis.

Chapter 2

Theoretical foundations

As already mentioned in the introduction, machine learning has become the primary tool for solving computer vision problems. The main objective of a machine learning process is to generate an algorithm to solve a certain task without being explicitly programmed by a human. It needs data to infer a mathematical model by optimizing a criterion or a loss function representing the goal of the task. Machine learning methods can be divided into the following categories:

- Supervised learning.
- Unsupervised learning.
- Reinforcement learning.

The main difference is in the formulation of the goal. Unsupervised learning methods do not have an annotated "ground truth" output that needs to be learned. The goal is to discover clusters or groups of data with similar patterns. Reinforcement learning methods are trained using higher-level feedback based on rewards and punishments mimicking the human learning process. The formulation of the criterion function is, therefore, more flexible compared to a standard supervised learning and applicable to non-differentiable learnable parameters.

Most machine learning algorithms belong to the traditional supervised learning paradigm divided into classification and regression problems. Both versions need precise manual markings of the desired output which, for classification algorithms, belongs to a discrete distribution of a limited set of values. Regression outputs, on the other hand, represent a continuous numerical value within a certain range. It also entails differently formulated criterion functions which are optimized over the learning parameters during the training process.

In the rest of this chapter, the basic machine learning building blocks for the proposed face alignment and facial expression recognition methods will be introduced. We will start with decision trees which can be grouped into ensembles to improve their generalization abilities. Local binary features are derived from the trained ensembles of decision trees to create compact task-specific features extracted from the image. The described local binary features are used in

combination with neural networks, also introduced in this chapter. Finally, CNNs are described as a special case of neural networks currently dominating the computer vision field.

2.1 Decision trees

Trees are an abstract hierarchical data structure with many different uses in computer science. From a graph theory perspective, an ordered tree is considered a connected, acyclic, directed graph which implies that any two nodes in the tree are connected by exactly one path. A node in a tree is a structure that may contain a value or a condition and has zero or more child nodes. Any node that has no children is called a leaf node, while nodes that have children are called parent nodes. The topmost node is called the root node, and the depth of the tree is defined as the longest path from the root node to a leaf node.

Decision trees are commonly used in decision analysis as a support and visualization tool, however from the perspective of this thesis, a much more interesting use is in the form of a predictive model in statistics, data mining, and machine learning. The most common version used is binary decision trees, where each parent node has two child nodes with binary conditions. The models are trained in a supervised manner and can be used for both regression and classification problems. The binary conditions or tests at each node perform a hierarchical segmentation of the training data with leaf nodes providing the predicted output. The parameters of the node tests are selected to minimize a loss function representing the purity of the resulting data partitions.

For illustration purposes, we will explore the construction of a decision tree on an artificial regression problem. We can assume a data set $\mathbf{x} \in \mathbb{R}$ with targeted output ("ground truth") $\mathbf{y} \in \mathbb{R}$ produced in the following way:

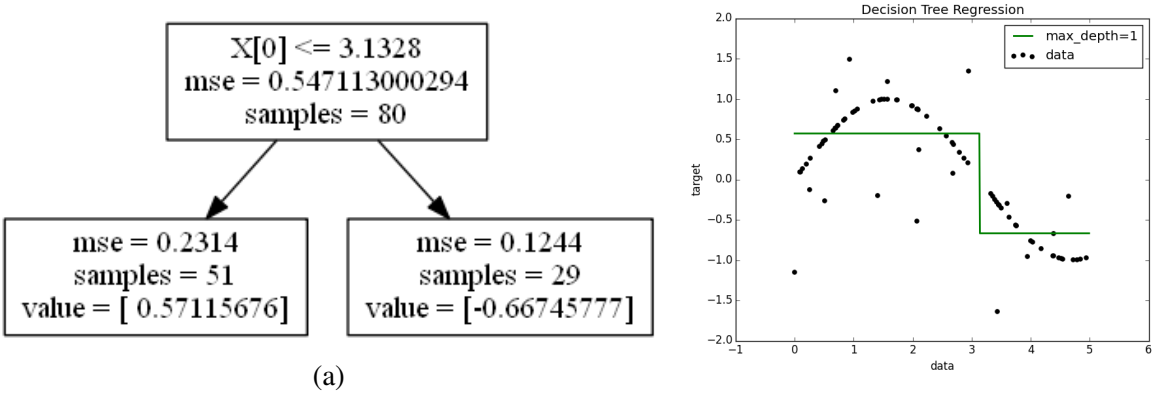
$$y = \sin(x) + \eta \tag{2.1}$$

where η represents the noise component. We can build a decision tree to predict y based on the input feature x. A simple binary test at each node can be formulated as follows:

$$T(x) = \begin{cases} 0, & x \leq t \\ 1, & x > t \end{cases} \tag{2.2}$$

where t represents a threshold selected from a pool of random values to minimize a chosen criterion or loss function. In this case, the most commonly used measure is the mean squared error (MSE) which is defined in the following way:

$$L = \sum_{y \in P_0} \frac{(\bar{y}_0 - y)^2}{N_{P_0}} + \sum_{y \in P_1} \frac{(\bar{y}_1 - y)^2}{N_{P_1}} \tag{2.3}$$



(a) (b)

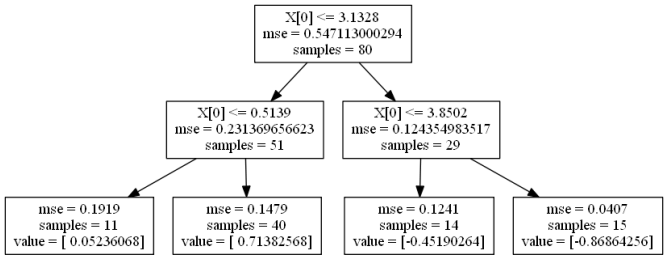
Figure 2.1: (a) Structure of the trained decision tree with a maximum depth of one producing (b) the output on a sinusoidal data set with noise.

where $\overline{y}_0$ and $\overline{y}_1$ represent the mean output values of partitions P_0 and P_1 produced by the binary test from Equation 2.2. The mean values in the leaf nodes represent the final output of the model. During the inference phase, the sample x_i is propagated through the tree and appropriate tests until it reaches a leaf node. The stored mean value from the corresponding training set partition of the leaf node is the predicted output $\hat{y}_i$ for the sample x_i.

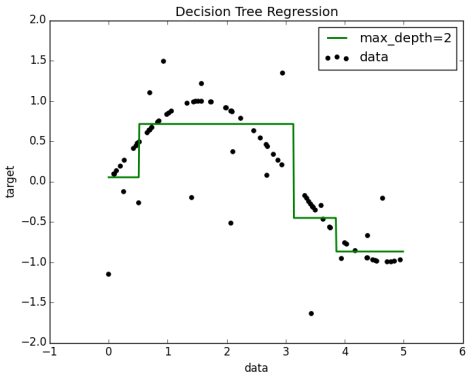
An example of a decision tree with a depth of one is presented in Figure 2.1a. The resulting prediction can be seen in Figure 2.1b using a data set produced by Equation 2.1. As can be seen, the data set consists of 80 samples following a sinusoidal curve with random outliers. The training process generated an optimal threshold value $t_0 = 3.1328$ at the root node, which splits the data set into two partitions with 51 and 29 samples, respectively. The average and, consequently, predicted values for the first and second partitions are $\overline{y}_0 = \hat{y}_0 = 0.5712$ and $\overline{y}_1 = \hat{y}_1 = -0.6675$, respectively. Such a small tree does not have the capacity to model this example data set with an MSE of $L = 0.1926$.

Growing an additional level through the training process improves the prediction ability of the decision tree, as can be seen in Figure 2.2. Two additional thresholds are introduced ($t_1 = 0.5139$ and $t_2 = 3.8502$), producing a total of four final partitions with corresponding average values as predictions and reducing the MSE to $L = 0.1297$. This nicely demonstrates the predictive power of decision trees.

It would be logical to add more nodes to improve the accuracy of the decision tree further. However, as the results in Figure 2.3 illustrate, this leads to over-fitting and learning random noisy elements of a specific data set. It degrades the generalization ability of the model and performance in real-world scenarios with different noise distributions. A simple way to combat this phenomenon is by enforcing a minimum size of the data partition for the node condition to be valid. As seen in Figure 2.3, there are five partitions with a single sample, all representing a noisy element of the underlying signal. Such a condition is, however, difficult to precisely

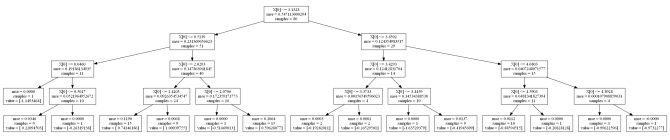


(a)

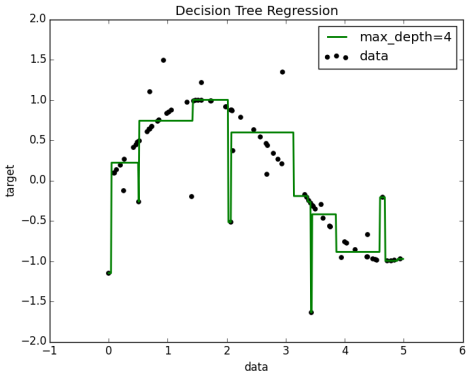


(b)

Figure 2.2: (a) Structure of the trained decision tree with a maximum depth of two producing (b) the output on a sinusoidal data set with noise.



(a)



(b)

Figure 2.3: (a) Structure of the trained decision tree with a maximum depth of four producing (b) the output on a sinusoidal data set with noise.

determine on different, complex data sets and rarely effective in battling over-fitting.

Instead of building more complex decision trees prone to high variance, another approach is to combine multiple smaller and weaker learners (by limiting their maximum depth) into a strong learner in the form of an ensemble. The two most common groups of techniques for forming ensembles are bagging and boosting. Bagging is a technique where each tree is trained on a randomly selected subset of the training data. A prominent representative of bagging methods is the random forest where, in addition to the randomized data, a random subset of features is used for each tree [16]. The final ensemble output is the averaged prediction of the individual trees in the forest. Boosting is a technique where each tree is trained sequentially, taking into account the errors of the previous tree in the succession. Gradient boosting is the most successful representative where the gradient descent algorithm is used to modify the sample weights to emphasize previously incorrect predictions [17]. In general, gradient boosting ensembles provide higher predictive capacity even though more prone to over-fitting when compared to random forests.

The advantages of decision trees as a machine learning tool are their simplicity and ease of use. Unlike other methods, decision trees can combine different types of features without normalization. The predictions can be easily interpreted by following the chain of decisions. Their non-linear nature provides high flexibility in modeling different data distributions. One of the main disadvantages is, as already highlighted, their vulnerability to over-fitting, which requires complex techniques and hyper-parameter tuning to overcome.

2.1.1 Local binary features

One drawback of the described ensemble methods, especially random forests, is the fact that each tree is trained independently and greedily to solve its local subset of the problem. The results are then averaged without mutual awareness, missing the potential synergy of complementary information. Ren et al. noticed this problem in [18] and offered their solution in the form of a global refinement procedure. Their method will be summarized here since it is an important prior work for this thesis.

To fully understand the proposed optimization, the tree prediction needs to be formulated differently. Two new terms will be introduced: the indicator vector $\phi(x)$ and the leaf matrix ω. The indicator vector represents the tree structure and can be considered a function that maps the input x to the corresponding leaf node. The leaf nodes are represented by a position in the binary vector resulting from the mapping function $\phi(x)$. If the sample x belongs to a leaf node l after traversing the tree, the corresponding position l in the binary vector will contain a binary 1, otherwise 0. If we populate the leaf matrix ω with local leaf predictions, the final prediction

of the tree $\hat{y}$ can then be formulated in a simple and compact way:

$$\hat{y} = \omega\,\phi(x) \tag{2.4}$$

The local leaf predictions populating the leaf matrix ω are usually mean values for regression and posterior probability for classification of the corresponding data partitions. It is a standard tree construction procedure as described in the previous section.

The objective function of a single tree can now be formulated in the following way:

$$\min_{\phi,\omega} \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y}_i) \tag{2.5}$$

The MSE is commonly used as the loss function L for regression problems as defined in the previous section in equation 2.3. For classification, the Gini impurity measure is usually used. More details will be presented in chapter 5 when applying it to facial expression classification. This formulation can be extended in a similar way to represent the objective of a random forest:

$$\min_{\phi,\omega} \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{N} L(y_i, \hat{y}_i^t) \tag{2.6}$$

where we now have T trees in the forest and $\hat{y}_i^t$ represents the prediction of the t-th tree for the i-th sample.

We know that the indicator vector $\phi(x)$ represents the tree structure (thresholds in binary node tests) built by taking into account the loss function. The leaf matrix ω, which represents the actual predictions however, is calculated on a local leaf subset of the data without regard of the actual objective in the training process. The minimization of the loss in equation 2.6 with respect to the leaf matrix ω is, therefore, suboptimal and can be improved using a global refinement of the leaf predictions. This refinement takes the following form:

$$\min_{\Omega} \frac{1}{2}||\Omega||_F^2 + \frac{C}{N} \sum_{i=1}^{N} L(y_i, \hat{y}_i) \tag{2.7}$$

where Ω and $\Phi(x)$ represent the concatenated parameters of all trees in the forest:

$$\Phi(x) = [\phi_1(x); \ldots; \phi_T(x)], \tag{2.8}$$

$$\Omega = [\omega_1; \ldots; \omega_T] \tag{2.9}$$

and the resulting prediction $\hat{y}$ is then:

$$\hat{y} = \Omega\,\Phi(x) \tag{2.10}$$

In practice, the ensemble trees are trained regularly as usual, which produces an optimal mapping function $\Phi(x)$. The local leaf predictions are then discarded and jointly refined according to the equation 2.7. This optimization can be efficiently solved using a support vector machine (SVM) to produce optimal leaf weights Ω for the objective function.

In the context of computer vision, decision trees can use a very efficient and simple binary test called pixel difference feature (PDF). It comprises a simple comparison of intensity values on two positions in an image as follows:

$$T(p_1, p_2) = \begin{cases} 0, & \mathbf{I}(p_1) \leq \mathbf{I}(p_2) \\ 1, & \mathbf{I}(p_1) > \mathbf{I}(p_2) \end{cases} \tag{2.11}$$

where $\mathbf{I}(p)$ represents the intensity value at pixel position p. Coupled with the described tree mapping function $\Phi(x)$ and using an image as the input $x = \mathbf{I}(p)$, this method can be considered a task-specific feature extraction producing local binary features. Looking at the global refinement procedure from this perspective, the predictive power can be further improved by using artificial neural networks instead of an SVM. More details about these improvements are presented on specific problems in chapters 4 and 5.

2.2 Artificial neural networks (ANNs)

Artificial neural networks are computing models loosely inspired by biological nervous systems with first theoretical implementations dating back from the 1940s [19]. With the increase in computational power and parallel processing using graphical processing units (GPUs), the practical applicability has increased tremendously in the last decade. ANNs and their variants are currently the dominant tools in artificial intelligence and computer vision.

The basic components of ANNs are neurons (also called perceptrons) and their connections. From a graph theory perspective, these are simply nodes and edges forming a directed, weighted graph. Each node takes multiple inputs and produces a single output. The inputs can be features extracted from data or outputs of other nodes. The weighted sum of all inputs to a node forms its activation value passed through a nonlinear activation function to produce the final output. The nodes are usually grouped in successive layers based on the input/output topology. The final layer of nodes produces the prediction of the ANN model. Such simple, feedforward, and fully connected ANN architectures where each node in one layer is connected to every node in the next layer are usually called multi-layer perceptrons (MLPs). Figure 2.4 shows an example of an MLP topology.

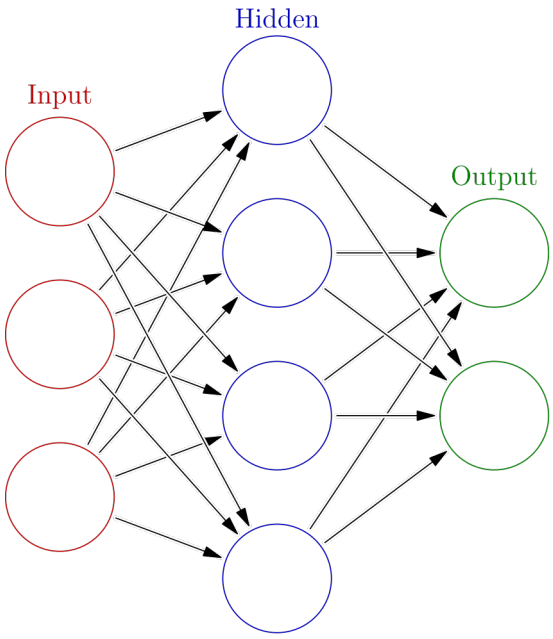


Figure 2.4: A simple MLP architecture with an input, hidden, and output layer.[a]

[a]Image Colored neural network.svg from author Glosser.ca, licensed under CC BY-SA 3.0.

2.2.1 Optimization algorithms

The central component of every ANN training in a supervised manner is the backpropagation algorithm [20]. It efficiently calculates the gradient of a loss function with respect to the ANN's weights utilizing the chain rule. The ANN is essentially a combination of composition functions and matrix multiplications. Since each layer's weights affect the loss only through the next layer, the backpropagation algorithm avoids duplicate calculations and unnecessary intermediate values by calculating the gradient backward from the last layer to the first. This, in turn, enables the use of gradient-based methods for optimization to minimize the loss and essentially teaches the model to predict specific outputs using annotated data sets.

If we have the objective function of the ANN defined as $L(\mathbf{W})$, the iterative step of the basic gradient descent then takes the following form:

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \gamma \nabla L(\mathbf{W}_n) \tag{2.12}$$

where $\mathbf{W}_n$ represents the ANN weights at iteration n, γ the step size, and $\nabla L(\mathbf{W}_n)$ the gradient of the objective function with respect to the current ANN weights. This produces a succession of weights from random initial values to the optimal weights with the minimal value of the objective function at iteration n:

$$L(\mathbf{W}_0) \geq L(\mathbf{W}_1) \geq \ldots \geq L(\mathbf{W}_n) \tag{2.13}$$

Such an iterative algorithm guarantees convergence to a local minimum and, for convex objective functions, the global minimum. It is, however, rarely the case for real-world machine learning problems with complex non-convex objective functions. A much more common situation is trapping the optimization process in a sub-optimal local minimum or even overshooting a global minimum. The selection of the step size and initial weight values [21] is, therefore, very important for the optimization process.

Many solutions for this problem have been proposed over the years. A natural extension of the basic algorithm described above is Newton's method, where the curvature of the objective function is inspected via the Hessian operator (the second derivative). The Hessian inverse can be used to calculate the optimal step size. However, it is computationally infeasible for large deep learning networks. Quasi-Newton methods avoid computing the Hessian through approximation. The most popular representative is the BFGS method [22] and the limited-memory version L-BFGS [23].

Speaking of infeasibility, even without the second derivative, gradient descent is computationally impractical on large data sets, which is why it is seldom used. Stochastic gradient descent (SGD) approximates the gradient using a randomly selected subset or batch of the training set, achieving faster iterations but slower convergence. It is currently the standard for ANN optimization. Due to the stochastic nature, the gradients tend to be noisy, making the step size (known as learning rate in machine learning) even more important. A common practice is to dynamically change the learning rate according to a predefined schedule with many different variants.

One such useful technique is the introduction of average gradients over past iterations of SGD. The momentum parameter determines the amount of influence of the current gradient to the accumulated average [24, 25]. This technique also helps to stabilize the convergence of the optimization. However, it introduces an additional training hyper-parameter. In practice, many objective functions exhibit uneven structures where some feature dimensions greatly outweigh others, making a single learning rate schedule ineffective. In the Adagrad optimization algorithm, this problem is successfully addressed by adjusting the learning rate for each dimension separately using the aggregated magnitude of previous gradients [26]. Since the magnitude is constantly growing, the learning rate is diminishing quite aggressively, which is improved in the RMSProp algorithm [27]. The learning rate schedule is decoupled from the dimension adaptivity by introducing momentum to the gradient magnitude aggregation. Another variant of Adagrad is Adadelta which decreases the amount of adaptiveness to different dimensions [28]. An additional benefit is the learning rate parameter elimination by utilizing the amount of change itself as the calibration for future change. The currently most popular optimization method for deep learning, combining all of the above-described techniques, is the Adam method [29]. It uses weighted averages for both momentum and scale combined with an explicit learn-

ing rate parameter to address potential issues with convergence resulting in a robust and efficient optimization algorithm.

2.2.2 Convolutional neural networks (CNNs)

In addition to different optimization algorithms, much effort has been invested in ANN architecture research. Instead of using a simple, stateless feedforward structure, recurrent neural networks (RNNs) use a feedback loop and internal memory with great success on problems with variable-length sequences of inputs [30]. Another architectural breakthrough is generative adversarial networks (GANs) using the competition of multiple networks in training to generate new information [31]. Since architecture design is a tedious process, a new field emerged to automate this using neural architecture search (NAS) and meta-learning [32]. The most important architectural invention from the perspective of computer vision is the convolutional neural networks.

Similar to ANNs, CNNs are inspired by the visual cortex of the brain. Due to the high dimensionality of images as inputs, a naive application of ANNs leads to an explosion of parameters even with the smallest of resolutions. Instead of a fully connected layer to each pixel as input, convolutional layers use filters with shared weights shifted step-by-step and applied across the image to produce their output [33, 34]. This reduction in the number of parameters allows for longer chains of convolutional layers to produce deep architectures and the emergence of deep learning. Another benefit of such architectures is the data-driven adaptation of filters producing both low-level (edges or blobs) and high-level features (relevant abstractions) throughout the deep chain of layers. This proved to be superior to hand-crafted features traditionally used in image processing. In combination with parallel processing on GPUs, deep learning started its domination in computer vision over the last decade.

The most important component of a CNN architecture is the convolutional layer. As already mentioned, it consists of filters or kernels which are convolved with the input (the image or the activation map of the previous layer). The kernels' weights are modified (optimized) during the training process, while the following hyper-parameters are determined beforehand during the architecture design:

- Size of the kernel - the filters are usually squares with the same width and height (e.g., 3x3 or 5x5).
- Depth of the kernel - represents the number of different 2D filters in the layer.
- Stride - controls the step size while sliding and convolving over the input.
- Padding - determines the number of additional rows and columns added to the input (usually filled with zero values).

The above-described parameters of the convolutional layer determine its receptive field - the local region of the input affecting the resulting feature map. They also form the size of the map

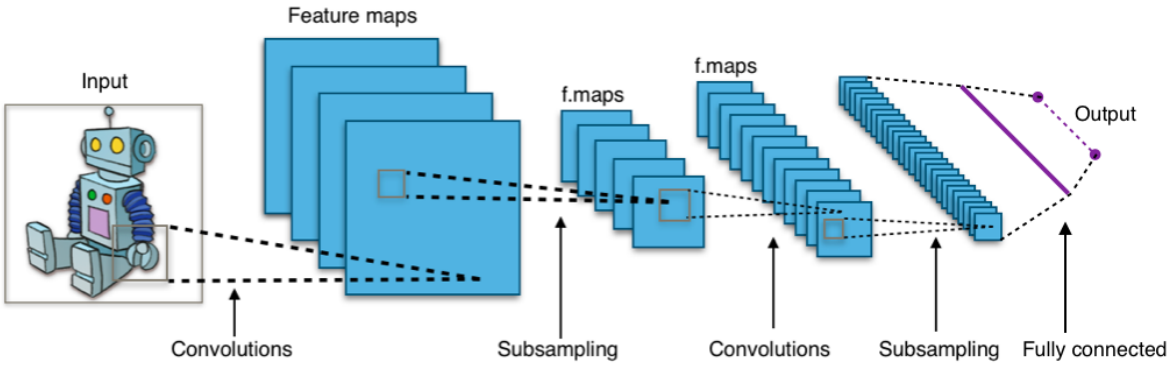


Figure 2.5: An example of a classical CNN architecture similar to LeNet-5 [35] by LeCun et al.[a]

[a]Image Typical cnn.png from author Aphex34, licensed under CC BY-SA 4.0.

giving a down-sampling or up-sampling functionality to the layer.

Another important layer in CNN architectures is the pooling layer, an additional method for non-linear input down-sampling. It divides the input into non-overlapping regions and outputs a single value for that region. The most common variations are the max and average pooling layers, which produce the maximum and average value of each region, respectively. Along with other down-sampling techniques, it reduces the size of the feature maps and associated parameters, improving both computational efficiency and generalization ability (preventing over-fitting). Additionally, it improves the translational invariance of the CNN architecture. Finally, after sufficient down-sampling, fully connected layers are added for final classification or regression. An example of a standard CNN architecture is shown in Figure 2.5.

The first classical CNN architecture called LeNet was proposed by LeCun et al. in the 1990s for handwritten character recognition [35]. Since computing power and large-scale data were very limited at the time, there was little progress up to the 2010s. With the rise of GPUs as general-purpose computing tools and the availability of cheap cameras and mobile phones allowing big data, the door was open for deep learning to dominate computer vision. The first large-scale success occurred on the ImageNet challenge for general object detection and recognition with millions of images of thousands of classes of objects [36]. Krizhevsky et al. proposed a much larger and deeper CNN architecture than LeNet named AlexNet, winning the ImageNet challenge by a large margin [37]. Using an efficient GPU implementation of the convolutional layer, the authors improved the training time by an order of magnitude, which allowed more data to be used. It set the tone for the next decade leading to different architectural variations. VGG networks use stacks of smaller 3x3 convolutions instead of large ones used in AlexNet, resulting in larger learning capacity but also higher over-fitting risks [38]. To create even deeper networks, He et al. introduced a residual block which adds a direct skip connection after each two successive convolutional layers [39]. It solves the vanishing gradient problem allowing for very deep architectures with hundreds of layers. Around the same time, another

significant improvement was proposed by Szegedy et al., introducing a bottleneck block in their first of Inception architectures called GoogLeNet [40]. The main building block is the Inception module which combines kernels of different sizes in parallel. However, it employs dimension reduction using 1x1 convolutions before expensive 3x3 and 5x5 convolutions to improve execution time. The described methods are just highlights from the very active research field of deep learning in computer vision.

Chapter 3

Related work

In this chapter, we will introduce relevant prior work for both face alignment and facial expression recognition. Since face alignment is a prerequisite for many other face-related problems, it has received wider attention from the research community and requires a more thorough overview [41] than facial expression recognition. An overview of various regression architectures is given in section 3.1, highlighting the strengths and weaknesses of each approach. In section 3.2, 3D alignment methods are introduced and analyzed. Multi-task learning, a relatively new technique, is investigated in section 3.3 in the context of face analysis. Another important topic for face alignment is described in Section 3.4, exploring how to handle partial occlusions of the face. FER-related work covering hand-crafted features, feature fusion, and deep learning methods are presented in section 3.5.

3.1 Face alignment regression architectures

Regression-based or, as they are also often called, discriminative methods estimate landmark positions directly from facial images. These methods demonstrated superior accuracy, speed, and robustness compared to earlier, traditional methods that involve Active Appearance Models [42, 43], Active Shape Models [44], and local part classification using search algorithms. Such constructed models demonstrate poor ability to express all combinations of face variations due to expressions, illumination, and head pose [45]. A standard regression problem formulation is commonly used where the target values are difference vectors between an initial shape estimate and the ground truth shape using features extracted from images. The initial shape estimate is usually a mean shape calculated from the training set normalized to the ground truth bounding box.

Earlier methods used regression for each landmark individually based on the local appearance around the initial position and additionally enforced a global shape constraint to make the local estimations more robust. These methods are described in more detail in section 3.1.1.

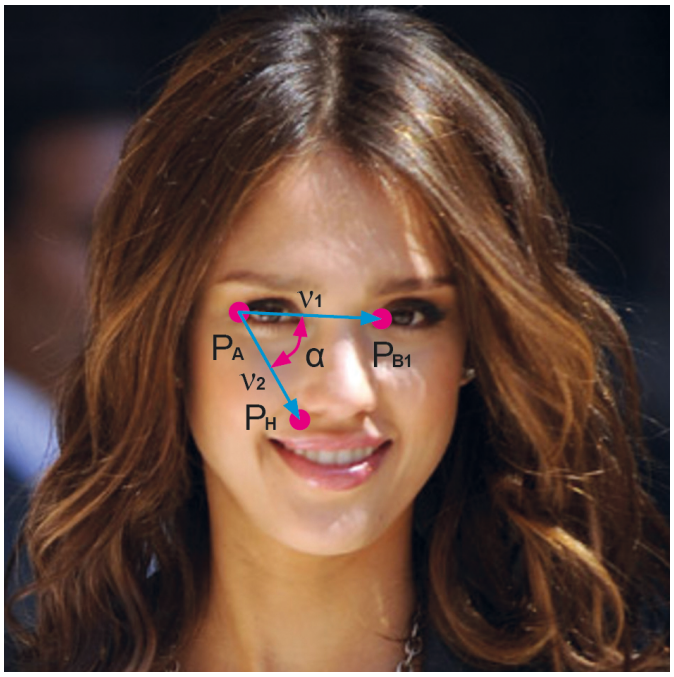


Figure 3.1: The relationship between landmarks as modeled in the BoRMaN method [46].

Later on, researchers used the joint training process for all landmarks to create an implicit shape constraint making the methods more straight-forward and simple to train. The innovation came in pair with a new cascaded architecture which breaks up the problem and solves it in a coarse-to-fine manner. This cascaded architecture achieved good results and has been further developed into many variants which are systematically covered in sections 3.1.2, 3.1.3, and 3.1.4. Finally, as in many other computer vision fields, researchers investigated deep learning methods and CNNs for face alignment both in a pure end-to-end and cascaded regression framework (section 3.1.5).

3.1.1 Constrained Regression

Constrained regression methods learn to induce individual landmark positions directly from image features but employ a corrective step that ensures global face shape constraint.

A representative algorithm from this group is Boosted Regression with Markov Networks (BoRMaN) [46]. Support Vector Regression (SVR) with a Gaussian Radial Basis Function (RBF) kernel is used as a local regressor for each landmark. The method uses Haar filter responses as features. An initial estimate of landmark locations is a mean shape placed relative to a bounding box returned by a face detector. Each prediction is then refined using Markov Random Fields (MRF) that model the global relations between landmarks. Each node in the model is a vector between two landmarks. The relation between two nodes is modeled as the difference of angles and the ratio of the lengths of these vectors (Figure 3.1). It ensures robustness to scale, rotation, and translation variations. The positions of the landmarks are iteratively

updated. At each iteration, the Markov Network analyzes the current predictions and generates the sampling regions for the next iteration. In the process, stable points are used to aid in the prediction of non-stable landmarks.

Kazemi and Sullivan in [47] proposed a method that uses a sliding window approach to detect the face parts (mouth, nose, and eyes) within the previously detected face region with a constructed tree structure to enforce shape constraints. After the parts are located, individual landmark linear regressors are used on the image patches to find the landmark points of the corresponding parts. A variant of Pyramid of Histograms of Orientation Gradients (PHOG) feature descriptors described in [48] is used for both detectors and regressors.

A constraint technique similar to BoRMaN [46] is employed in [49]. The method is called Structured Output Regression Forest (SO-RF), and its spatial constraints are manually modeled by a structure graph. Each landmark has a directed graph associated with it that defines its influence on the neighboring landmarks. Each leaf node models the affiliation to a base landmark and stores an offset and a confidence. Additionally, each leaf node models the relative offsets to the neighboring landmarks with a Gaussian distribution. The combined votes from the local evidence and the spatial constraints form a map where the highest probability landmark position is found. In their later work (see [50]), Yang and Patras use the same regression forest voting scheme, but the shape constraints are replaced with sieves that act as filters for the votes. The forest leaves cast votes for the landmarks and face center simultaneously. A Hough map is formed from the votes, and the mean-shift algorithm is then used to find the maximum likelihood detections for the landmarks. The first sieve used is a face center sieve that discards the votes not consistent with the global face center hypothesis. The votes are then filtered by proximity threshold sieves, where the threshold is iteratively adapted based on the decision from a classifier trained on features extracted from Hough maps.

The final representative method from this group is Local Evidence Aggregated Regression (LEAR) [51]. The overall idea of the method is to use predictions from local individual regressors and shape constraints as in BoRMaN [46] to update the sampling region in the next iteration. Additionally, each iteration prediction is accumulated into a probability map from which the final prediction is made. Local Binary Patterns (LBP) [52] extracted from patches are used as feature vectors and SVR to regress the offset vector. The regressors are trained to be precise as opposed to general by limiting the variance of the training set sampling locations. The outlier predictions thus produced in the inference phase are then mitigated by aggregation of all estimates from previous iterations. The regressor output is evaluated by performing another regression from the predicted location using the output distance to measure confidence.

The methods described in this section are the earliest attempts of robust face alignment (Table 3.1). It became evident that local landmark appearance, although very important, is not sufficient for accurate localization. Information from neighboring landmarks and global face

Table 3.1: Constrained regression methods summary.

Methods	Features	Regressor	Constraint	Year
BoRMaN [46]	Haar-like filters	SVR with Gaussian RBF	MRF	2010
Kazemi & Sullivan [47]	PHOG	Ridge	Tree structure	2011
SO-RF [49]	HOG & Gabor	Random forest	Structure graph	2012
Yang & Patras [50]	HOG & Gabor	Random forest	Sieves	2013
LEAR [51]	LBP	SVR	MRF sampling	2013

shape configuration is equally important in order to solve extreme variations in facial appearance. The first attempt to utilize the face shape is through constructed constraints and corrective post-processing after individual landmark localization. It was an important step in the right direction. However, it is very difficult to manually construct such a constraint to accommodate all possible variations and still provide the needed robustness. An additional weakness is the use of hand-crafted features for landmark localization that suffer from similar problems. As in other computer vision fields, a shift towards data-driven modeling occurred in face alignment as well.

3.1.2 Cascaded Regression

Cascaded regression has established itself as the leading approach for face alignment due to its speed, robustness, and accuracy. In this framework, several regressors $(R^1,\ldots,R^t,\ldots,R^T)$ are successively applied starting from the initial shape estimate S^0 (Figure 3.2). Given an image I, each regressor learns and estimates a shape increment δS and updates the face shape:

$$\delta S = R^t(I, S^{t-1}) \tag{3.1}$$

$$S^t = S^{t-1} + \delta S \tag{3.2}$$

where the tth regressor R^t updates the previous shape S^{t-1} to the new shape S^t [45]. It is important to note that the tth regressor depends on the previous shape estimate S^{t-1}. The dependency is usually through shape-indexed features, which is a concept first introduced in [53]. The method is called Cascaded Pose Regression (CPR) and was developed for general object alignment, including faces as well. The method owes its success to pose-indexed features where pixel positions used in the pixel difference features are stored relative to the object pose and are thus consistent across large pose variations. Random fern regressors were used at each stage of the cascade.

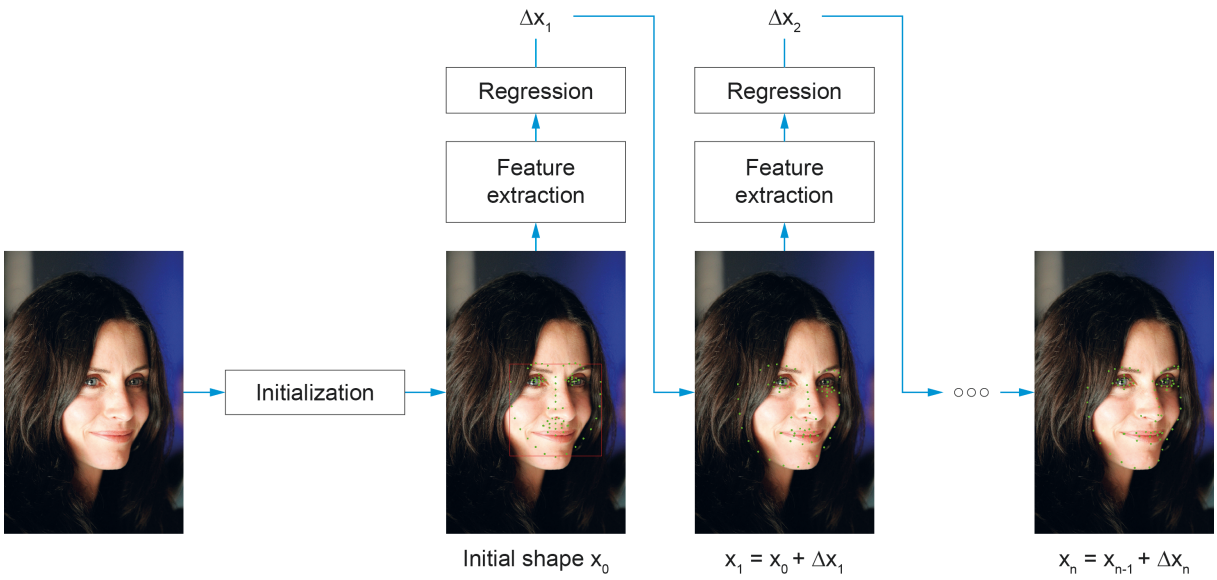
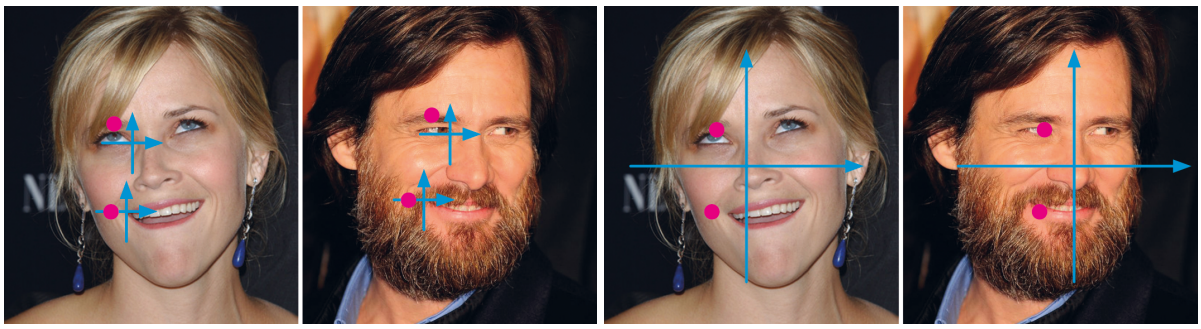


Figure 3.2: Cascaded regression in a coarse to fine manner.



(a) (b)

Figure 3.3: Shape-indexed pixel positions (a) and globally indexed pixel positions (b). Shape-indexed features retain same semantic meaning regardless of face variations [45].

Cao et al., in their seminal work called Explicit Shape Regression (ESR), extend the idea from CPR [45]. Again, pixel difference features and fern regressors are used. However, the shape is jointly regressed as a vector which implicitly enforces a shape constraint in a non-parametric way (Eq. 3.1). They use a two-level boosted regression where each regressor in the cascade uses global features indexed relative to the nearest landmark (see Figure 3.3). Each regressor in the cascade is also a cascade of primitive regressors (ferns) using fixed features. The authors used correlation-based feature selection when choosing the most discriminative features from the pool.

Kazemi and Sullivan, in their work Ensemble of Regression Trees (ERT), improve upon the ESR method [54]. Instead of random ferns, gradient tree boosting ensembles are used. They also use shape-indexed features indexed to the closest landmark. However, they transform the pixel positions to compensate for rotation, scale, and translation relative to the mean shape. A prior is introduced to favor closer pixel differences in their feature selection process. They

use weights in the training-node split-error calculations in order to handle uncertain/occluded landmarks (the ground truth of some landmarks can be "turned off" when optimizing).

A somewhat different approach in the same framework is described in [55]. The algorithm is called the Supervised Descent Method (SDM), and it presents shape-indexed features and cascaded regression as a Newton-type optimization of a non-linear least-squares problem. Basically, they use linear regression and local Scale-Invariant Feature Transform (SIFT) features from [56] on local patches centered on currently estimated landmark positions. Eq. (3.1) is then replaced by:

$$\delta S = W^t \phi^t (I, S^{t-1}) + b^t \tag{3.3}$$

where W^t and b^t are linear projection matrix and bias term, respectively. ϕ^t is a non-linear global feature extraction function that concatenates local features extracted around currently estimated landmarks.

A method called Local Binary Features (LBF) by Ren et al. [15] is an improvement of ESR [45] and SDM [55] methods. A random forest is used, trained to minimize the alignment error for individual landmarks to produce binary features. Local features are coded in a binary array by placing a one for leaves where the sample ends up while traversing the tree and zero otherwise. Features that are individually trained for each landmark are then concatenated into a global feature vector as input for ridge regression (linear regression with L_2 regularization). This method owes its success to a feature learning step where features are explicitly trained for the given custom task instead of manually crafted (such as SIFT). Due to the sparseness of the feature vectors, the inference phase can be reduced to traversing the forest and performing simple look-ups and additions. Ren et al. achieved a frame rate of 3000 FPS which is, of course, hardware-dependent but impressive, nevertheless. In a later work, Luo et al. modified the forest to obtain probability features and used Probabilistic Random Forests (PRF), which modeled the probability for each sample belonging to a tree leaf node [57]. It slowed the algorithm down considerably because every sample must traverse every tree branch and the produced features are no longer sparse binary vectors. However, improved accuracy and stability (reduced noise during tracking on videos) is achieved. Another extension was presented in [58]. The main idea is to replace ridge regression with a neural network architecture utilizing a bottleneck layer. By doing this, the authors improved the accuracy, execution time, and reduced memory requirements of the original algorithm.

A similar, recent fast cascaded regression approach called Cascade Gaussian Process Regression Trees (cGPRT) was proposed in [59] by Lee et al. Features that are computed as difference-of-Gaussian filter responses on local retinal patterns referenced by the shape estimates are used instead of standard shape-indexed pixel differences as in ESR [45], ERT [54], and LBF [15]. For regression, Gaussian processes with a kernel modeled by trees are used,

optimized for the individual landmarks. Both innovations improved the results from previous methods in [15, 45, 54].

Recurrent cascaded regression

An interesting modification to the standard cascaded regression approach was recently proposed in [60]. The authors argue that there is a loss of knowledge between independently trained stages in the standard cascaded approach and propose a single Recurrent Neural Network (RNN) architecture that combines the training of all stages through the introduction of a state vector that serves as a mnemonic unit. The approach extends the classical Supervised Descent Method [55] with the use of an RNN, as already mentioned, and, additionally, with local small CNNs as feature extractors instead of hand-crafted SIFT features. The authors conveniently named the method Mnemonic Descent Method (MDM) and showed that the introduced state vector partitions the training set into meaningful clusters with different descent directions in subsequent stages.

A similar approach was presented in [61]. Liu et al. use the same architecture. However, they investigate the correlation of neighboring landmarks to remove redundant information of overlapping patches. To this end, the mentioned correlation is explicitly modeled and utilized under a multi-task learning paradigm. Additionally, multi-scale images are used to enhance coarse-to-fine alignment through the use of an RNN.

A couple of major improvements were introduced with the adoption of the cascaded regression framework. The global shape information is no longer constructed by hand. It is implicitly deduced from the training set, which demonstrated greater generalization ability. Furthermore, landmarks are regressed jointly, not individually, utilizing both local features and contextual information from neighboring landmarks. Finally, the complexity of the face alignment is broken down into a series of simpler problems through the cascaded architecture. The early stages of the cascade naturally focus on rough alignment dealing with head pose and shape variations, while the later stages focus on local details and subtle variations in facial appearance.

In later developments, cascade stages are treated as a sequence of inputs which makes sense both from a practical and theoretical standpoint. A single model for all stages reduces memory requirements and retains knowledge between individual stages. Through the use of a state vector ("mnemonic") inside the RNN architecture, the model can be made aware of decisions from previous stages and hence learn conditionally based on those decisions. As an additional bonus, this method can be naturally adjusted to tracking from a video where information from previous frames can be efficiently utilized.

The summary of the described cascaded regression methods, with highlights of key differences, is presented in Table 3.2. These methods nearly saturate frontal and relatively constrained data sets. However, "in the wild" data sets are still challenging due to a weakness to initializa-

Table 3.2: Cascaded regression methods summary.

Methods	Initialization	Features	Regressor	Year
CPR [53]	Random	Pixel difference	Random ferns	2010
ESR [45]	Random	Pixel difference	2-level ferns	2012
SDM [55]	Mean shape	SIFT	Linear	2013[a]
ERT [54]	Mean shape	Pixel difference	2-level trees	2014
LBF [15]	Mean shape	LBF	Ridge	2014
PRF [57]	Mean shape	PF	Ridge	2015
cGPRT [59]	Mean shape	DoG	GPRT	2015
MDM [60]	Mean shape	Local convolutions	RNN	2016[b]
LBF-NN [58]	Mean shape	LBF	Neural network	2018

[a] `https://www.youtube.com/user/xiong828/videos`
[b] `http://trigeorgis.com/mdm`

tion that these methods demonstrate. Local shape-indexed features, frequently used within the framework, do not capture a large enough context for samples initially far away from the ground truth. These situations occur frequently in unconstrained data sets, including, e.g., full profile facial images. Strategies to address this issue within the framework will be described in the following sections.

3.1.3 Global feature initialization

Zhu et al. studied initialization strategies for face alignment and confirmed its importance for cascaded regression methods [62]. They managed to improve the robustness by initializing the ERT [54] cascaded regression method with an additional ERT [54] model trained on a subset of rigid landmarks to produce an improved initial shape. On the other hand, Valle et al. [63] addressed the problem of initialization sensitivity of the cascaded regression approach by introducing a CNN-based initialization stage. The method is called Deeply-initialized Coarse-to-Fine Ensemble (DCFE). A simple CNN architecture is used to estimate landmark probability maps using the whole face image as the input (see Figure 3.4). These initial landmarks are then utilized as input to a 3D model fitting procedure, which produces a robust and accurate initial face shape estimation. The fast ERT method [54] is used for precise alignment in the next stages of cascaded regression. The final stage uses separate models for facial regions in order to decouple the movements and achieve improved alignment for asymmetrical facial expressions.

A similar initialization approach was proposed in [64]. Again, the whole face image was

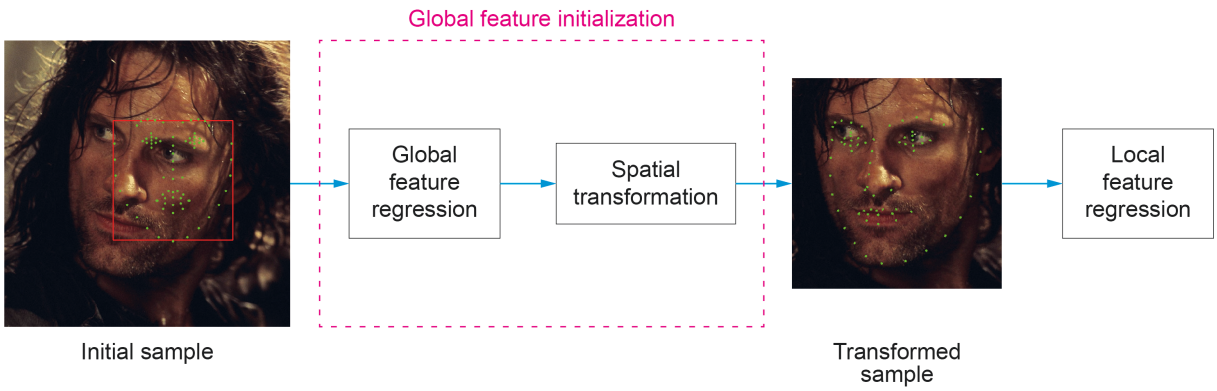


Figure 3.4: Global feature initialization - the main idea is to use the whole face region to estimate the initial face shape in order to calculate and eliminate the spatial transformation with respect to the canonical shape.

used for feature extraction in the first stage. However, Kowalski et al. used K-cluster Regression Forests with Weighted Splitting (KRFWS) to regress the 3D head pose parameters, which then served as initial landmark positions (after the projection of mapped vertices to the image space). Subsequent stages are designed according to standard cascaded regression framework with KRFWS algorithm and shape-indexed Pyramid HOG features from [65].

An interesting method that combines deep convolutional networks for feature extraction and cascaded regression has been proposed in [66]. The authors named it Deep Cascaded Regression (DCR). It comprises three modules. The first module is a convolutional/deconvolutional network which serves as a feature extractor for the other two modules. It produces a deconvolution layer of the same size as the input image. The second module performs an initialization search. It uses the last deconvolution layer with a fully connected layer to learn, for each landmark separately, the probabilities of each pixel belonging to that particular landmark. It also generates a number of representative shapes from the training set using k-means. The landmark probabilities are then used to find the closest shape as the initialization. The third module performs cascaded regression using the initialization shape. Linear regression is used with a fully connected layer on the features extracted from the module-one deconvolution layer around the currently estimated landmark positions (shape-indexed) on fixed-size patches.

A similar multiple module approach was proposed by Liu et al. in their work called Pose-insensitive Dual Sparse Constrained Cascade Regression (P-DSC-CR) [67]. They use a deep convolutional neural network to detect the initial five landmarks and estimate the head pose. Separate cascaded regressors for each pose (frontal, profile) are trained. Cascaded regression is improved by adding dual sparse constraints. At each stage, landmark updates are first trained by Lasso regression which produces a sparse projection matrix. Then, the updated landmark positions are fitted to the sparse shape dictionary, which produces the estimate for the current stage and is the input for the next stage. The dictionary is constructed using K-SVD [68]

Table 3.3: Global feature initialization summary.

Methods	Global features	Global regression	Spatial transform	Local cascaded regression	Year
DCR [66]	Convolutional	Encoder-decoder	Initialization search	Deconv. features + linear	2015
KRFWS [64]	PHOG	KRFWS	3D-APR	LBF [15] + KRFWS	2016
P-DSC-CR [67]	Convolutional	CNN	In-plane rotation	HOG + Lasso	2016
DCFE [63]	Convolutional	CNN	POSIT	ERT [54]	2018

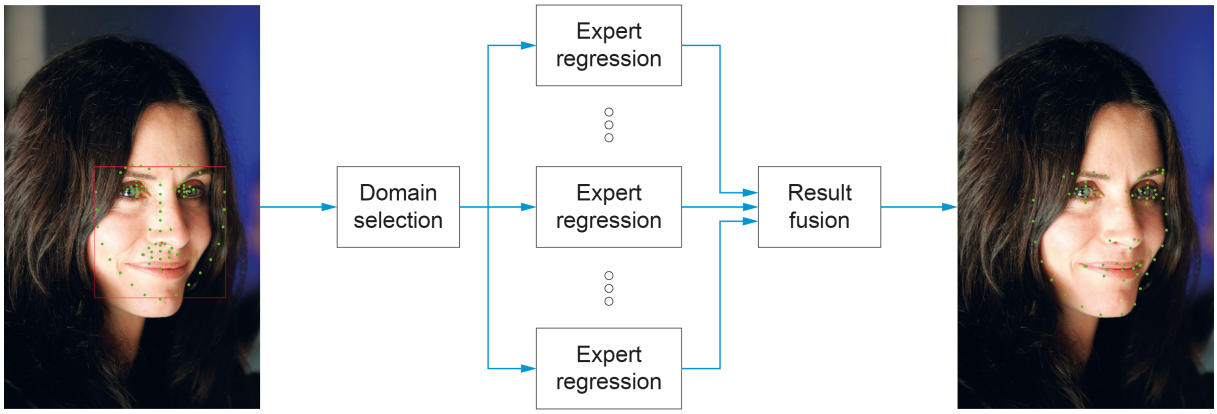


Figure 3.5: The cascade of experts can be divided into three elements: domain selection, expert regression, and result fusion.

algorithm on training faces. Multi-scale Histogram of Oriented Gradients (HOG) [69] features centered on landmark positions are used for regression.

In general, this approach can be graphically summarized as shown in Figure 3.4. Using an initialization stage with features from the whole face proved to be adequate to mitigate the initialization sensitivity problem of the cascaded regression approach. The initial stage takes a larger context around the initial mean shape as input which makes it robust to larger translation shifts from the ground truth face. Additionally, more complex algorithms with larger capacity (CNN) are usually used in the initial stage because of large input variance, while faster and more efficient features can be used in later stages to improve processing time and retain high accuracy levels. The described methods are summarized in Table 3.3.

3.1.4 Cascade of experts

The greatest face shape variations come from different head poses with respect to the camera. A straightforward way to improve face alignment accuracy is to use multiple domain-expert models in parallel in order to make the cascaded regression approach more robust to various head poses (Figure 3.5).

A simple parallel cascaded regression approach was proposed in [70] by Feng et al. called

Random Cascaded Regression Copse (R-CR-C). Three parallel cascaded regression threads are trained on random subsets of the training set and used at the inference phase. The regressors are plain ridge regressors using the Sparse Auto-Encoder features. A very similar approach was proposed in [71] using the FEC-CNN architecture [72] (described in Section 3.1.5) as the backbone. The method achieved very good results on the recent Menpo Challenge [73].

A more elaborate method was proposed by Xiong and De la Torre in their work Global Supervised Descent Method (GSDM), where they extend the original SDM [55] method to handle large pose variations [74]. The problem is again cast in a non-linear optimization framework where the aim is to find a globally better minimum by partitioning the optimization domain into regions of similar gradient descent. Mathematical theory is demonstrated that ensures such partitions exist and a procedure on how to find them. A separate SDM [55] model is trained for each region. The solution was applied to face tracking in videos.

Zhu et al., in their work Cascaded Compositional Learning (CCL) [75], developed a similar idea. Again, the optimization space is divided into multiple domains of homogeneous descent and separate experts trained for each domain. However, Zhu et al. added an explicit module to handle the initial domain selection instead of relying on the previous frame output as in the GSDM method. It makes the method more effective on images. The outputs of the individual experts are combined in a learning framework that directly optimizes landmark positions.

Dong et al., Zhu et al., and Rampal et al. have similar ideas on how to handle occlusions and extreme poses in their respective works Robust Discriminative Regression (RDR), Ensemble of Model Recommendation Trees (EMRT), and Ranked Parts Based Models (RPBM) [76, 77, 78]. The idea is to train multiple cascaded regressors using a heuristically determined subset of landmarks when extracting shape-indexed features. They all used linear regression and SIFT/HOG features. The difference is in how their output is combined at each stage. Dong et al. and Zhu et al. both use recommendation trees to learn weights used for the linear combination of estimates (quadratic programming is used to find the optimal weights at each node). Rampal et al. train a Support Vector Machine (SVM) to produce a ranking for each model using shape-indexed HOG features.

Finally, a probabilistic approach was proposed by Zhu et al. called Coarse-to-Fine Shape Searching (CFSS) method [79]. The main contribution is to search a shape sub-space at each stage in a coarse-to-fine manner from which initial shapes are sampled for regression. First, a shape library is created using Procrustes analysis. At each stage, the goal is to find a finer shape sub-space represented by a sub-space center and a more narrow normal shape probability distribution. The initial probability distribution is uniform representing equal chances for every shape in the library to be selected. Several initial shapes are sampled using the estimated shape probability distribution, and the regression is performed for each. The regressor outputs are combined using weights obtained through the dominant set approach and form the sub-space

Table 3.4: Cascade of experts summary.

Methods	Domain selection	Expert regression	Result fusion	Year
R-CR-C [70]	Random	Sparse Auto-Encoder + Ridge	Average	2015
GSDM [74]	Homogeneous descent	SDM [55]	Single result	2015[a]
CCL [75]	Homogeneous descent	LBF [15]	Composition ridge regression	2016[b]
RDR [76]	Facial region	SIFT + linear	Learned weighted average	2015
EMRT [77]	Head pose & occlusion	SIFT + SVM	EMRT	2015
RPBM [78]	Facial region	HOG + linear	SVM ranking	2015
CFSS [79]	Shape sub-space distribution	BRIEF/SIFT + linear	Single result	2015[c]

[a] `http://goo.gl/EGiUFV`
[b] `http://mmlab.ie.cuhk.edu.hk/projects/compositional.html`
[c] `http://mmlab.ie.cuhk.edu.hk/projects/CFSS.html`

center. The probability distribution for the next stage is also estimated using the sub-space center. At the last stage, the sub-space center is the final estimate. At each stage, a cascade of linear regressors using either BRIEF [80] or SIFT [56] features (accuracy vs speed trade-off) is used.

Cascade-of-experts is a logical approach to reducing complexity by dividing the problem into sub-domains as illustrated in Figure 3.5. However, it comes with a greater computational cost since multiple models are trained and then used at the inference phase. The accuracy boost is evident in the respective papers but often comes with a cost of doubling or even tripling inference time and memory requirements. It makes the approach impractical in many scenarios. The summary of the described methods is presented in Table 3.4.

3.1.5 Deep Learning

Deep learning methods have recently gained popularity due to the advances both in hardware and optimization techniques. They have been applied in many computer vision fields, including face alignment as well.

However, there has not been much success with simple deep CNN architecture training to accurately locate landmarks on a face image. One of the reasons is the need for large data sets to make such an approach successful. Wu et al. proposed to unify data sets with different annotations to increase both the size and the variance of the training set [81]. An architecture called Deep Variation Leveraging Network (DVLN) was used, consisting of two CNN networks: Dataset-Across Network (DA-Net) and Candidate-Decision Network (CD-Net). The DA-Net was trained on the unified training set where the deep layers of the network were shared across sub-sets with different annotations, while the last fully connected layers were specific for each annotation configuration. Additionally, they normalized the data sets so that a single profile

view is present, reducing the complexity of the problem. The CD-Net was trained to recognize the view of the facial image (left or right profile) and select the correct output of the DA-Net which takes as input normal and flipped images. The method achieved the second-best result on the Menpo Challenge [73].

Another direct regression approach using deep learning was proposed in [82] where a doubly CNN architecture [83] was used, which is computationally more efficient than regular convolutions along with Fourier feature pooling to build strong holistic representations. In order to encode landmark correlation, the authors designed a layer with linear low-rank learning instead of a fully connected layer as the output layer.

An interesting idea was explored by Shao et al. using a deep learning model named Multi-Center Network (MCNet) [84]. A CNN architecture based on VGGNet was trained in a standard way for face alignment. However, the authors used that pre-trained model and its shared deep features to separately fine-tune seven landmark regions, improving the precision of the original model.

Deep cascaded regression

Since cascaded regression achieved breakthrough results for face alignment, the logical next step was to combine it with deep learning. Sun et al. were pioneers in this area with their work called Deep Convolutional Network Cascade (DCNC) and proposed a cascaded regression approach with three stages of convolutional networks [85]. A shape with only five landmarks is estimated. Predictions from multiple networks are fused together at each stage to improve the accuracy and reliability of the estimation. The first stage networks take the whole face image as input and predict initial estimates of the landmark positions. The next two stages use patches centered on the estimated landmark positions as input from the previous stage and refine the estimations to achieve higher accuracy.

In a similar work called Coarse-to-Fine Convolutional Network Cascade (CF-CNC), Zhou et al. proposed to separate the detection process for inner and contour points [86]. The first stage neural network estimates the bounding boxes for inner and contour points separately. The second stage gives an initial holistic prediction of inner and contour points, also separately. The third stage refines the six facial parts, computes their rotations, and normalizes them before giving the patches to the fourth and last stage to make final refinements. The contour points do not utilize the third and fourth stages. Zhang et al. use a similar framework with stacked auto-encoder networks in their work named Coarse-to-Fine Auto-encoder Networks (CFAN) [87].

Kowalski et al. combined deep learning networks with the cascaded regression framework in [88] naming the method Deep Alignment Network (DAN). All stages in the cascade use the global facial region as input to the deep convolutional networks. In order to keep the advantages of shape-indexed features and transfer of knowledge between stages, the authors implemented

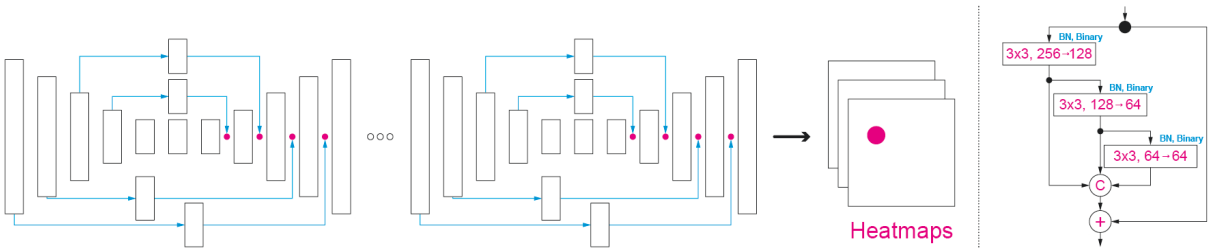


Figure 3.6: The hourglass CNN architecture and residual block used in [92].

connection layers that generate inputs for the next stage based on the output of the current stage. According to their experiments, two stages were enough to achieve convergence. A similar two-stage framework was used in [89]. However, the authors investigated the optimal loss for landmark regression, arguing that the universally used L2 loss is too sensitive to outliers in the data set. Through intuition and experiments, they derived a formulation of the Wing loss, which can balance the influence of both small and large errors during training. It enabled them to use relatively simple CNN architectures in both stages while achieving competitive accuracy.

All of the above-mentioned methods had their stages in the cascade trained individually and separately from the others. In [72], the authors claim that the cascade could benefit from joint training of the stages enabling the flow of information between them. A Fully End-to-End Cascaded CNN (FEC-CNN) architecture is introduced, which uses local shape-indexed CNN features in each stage and is optimized jointly using Stochastic Gradient Descent (SGD) and back-propagation. The biggest challenge was to generate gradients of shape-indexed patches of the image with respect to the input shape from the previous stage. They managed to successfully formulate the derivations by drawing inspiration from Spatial Transformer Networks [90]. The experimental section confirmed the benefits of the end-to-end training procedure. In a later work, Dapogny et al. strove for a similar goal of end-to-end optimization in their work Deep Convolutional Cascade for Face Alignment (DeCaFA) [91]. This was achieved, however, using fully convolutional stages with U-net architecture and transfer layers designed to produce landmark-wise attention maps. In order to use heterogeneous data (multiple data sets with different annotations), they used chaining of multiple transfer layers ordered by the density of landmarks in the face shape.

Heatmap regression

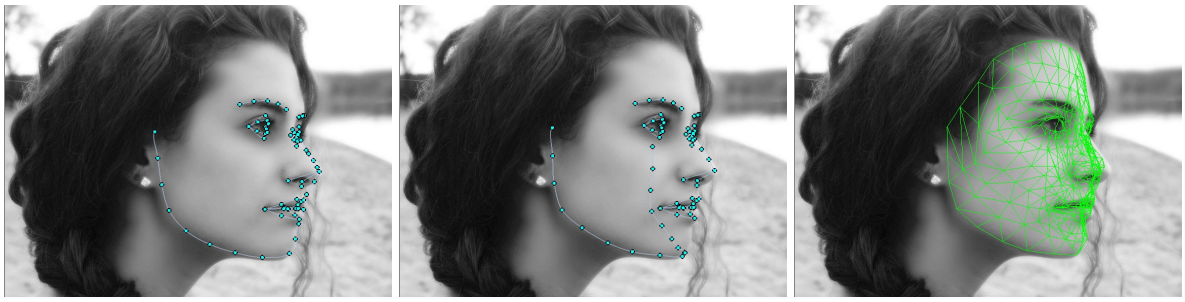
A recent method in [92] reached a saturation performance by using a modern deep CNN architecture and a generated, large data set, making the face alignment problem solved in most scenarios. The authors used a state-of-the-art hourglass CNN architecture with a novel residual block (see Figure 3.6) and trained each landmark's location as a heatmap which produced estimates on position certainty as well. The same network architecture was trained to convert 2D landmark annotations to 3D and create a large-scale 3D facial landmark data set with ap-

proximately 230k images. A similar approach was proposed in [93] achieving state-of-the-art results in the recent Menpo Challenge [73]. The authors added a supervised face transformation step which eliminates rigid face transformations based on the output of a face detector and its reduced subset of detected landmarks [94]. It improved the method robustness and reduced the complexity of the problem for the stacked hourglass CNN training.

The success of both methods inspired other researchers to investigate the heatmap regression approach in greater detail using the same stacked hourglass architecture [95, 96, 97, 98]. An interesting improvement was proposed by Wu et al. in their work named Look at Boundary (LAB) [95]. The main premise is that most landmarks in the face shape are ill-defined, even in a frontal pose. Thus, they introduce face boundaries as a more suitable face geometry representation. The stacked hourglass architecture is, therefore, used to estimate high-quality boundary heatmaps using adversarial learning. The boundary heatmaps are then driving the regression CNN to produce accurate landmark positions. An additional benefit of the boundary paradigm is the innate ability to represent heterogeneous annotations enabling the architecture to use a large, unified data set for training. Instead of using an additional CNN to produce landmark positions, Wang et al. estimate both landmark and boundary heatmaps using the stacked hourglass architecture [98]. However, their main contribution is modifying the Wing loss introduced in [89] and applying it to heatmap regression. Their Adaptive Wing (AWing) loss is designed to be more sensitive to small errors in the foreground and less on background pixels confirming the inferiority of the L2 loss once again.

Following similar reasoning, Liu et al. also stress semantic ambiguity of contour landmarks [96]. Instead of introducing boundaries, they opted for a probabilistic model of the "real" ground-truth. Landmark updates during training iterations are used to discern between random movements due to annotation noise and meaningful movements towards ground-truth. The probabilistically modeled "real" ground truth is then used in later iterations to achieve stable and more accurate convergence. Chen et al. addressed the same problem using Kernel Density Deep Neural Network (KDN) [97]. Instead of assuming a Gaussian distribution of the heatmap regression, their model can estimate a more general probability distribution, e.g., multimodal or asymmetric distribution. Furthermore, they extend the stacked hourglass architecture inspired by cascaded regression to propagate the estimated probability distribution between stages.

While all of the above methods aim to produce a low-resolution representation from which the landmarks are predicted, Wang et al. argue that high-resolution representation is beneficial for all spatial vision tasks [99]. In their novel HRNet architecture, parallel high-to-low convolutions are employed with a multi-resolution fusion scheme to exchange information across resolutions. The validity of their hypothesis and superiority of their architecture design is demonstrated on a wide range of vision problems: human pose estimation, semantic segmentation, object detection, and face alignment.



(a) (b) (c)

Figure 3.7: Different regression targets: standard sparse 2D landmarks (a), sparse 3D landmarks (b), and dense 3D landmarks (c). The largest difference between 2D and 3D landmarks can be observed on the contour landmarks.

Judging by the lack of successful research with a single, simple deep learning model for face alignment, it seems that the problem is too complex and data sets too small for such a straightforward approach. Thus, complex architectural and training procedures need to be implemented to achieve competitive results. The combination of deep learning models and cascaded architecture, where the problem is broken down into manageable sub-problems, is a promising solution. Similar to the group of methods from Section 3.1.3, the first stage in the deep regression cascade uses the whole face region to predict a subset of landmark positions. However, later stages also utilize CNN architectures with different techniques to focus the network on finer details. Methods from this group achieve great results in the wild. It comes with a greater computational cost though, since demanding CNN architectures are used throughout the cascade. Real-time performance is possible with the use of a modern GPU.

Another promising deep learning direction is the use of fully convolutional networks for heatmap regression with the widespread adoption of the stacked hourglass architecture. These methods achieve state-of-the-art results across different benchmarks. However, real-time performance is not possible even with a high-end GPU. It is understandable since an additional decoder block is necessary for the CNN architecture to estimate each pixel heatmap value. The summary of the described methods is presented in Table 3.5.

3.2 3D face alignment

Due to the ambiguity and self-occlusion of 2D landmarks in more extreme poses, 3D landmark alignment has gained traction in recent years. The ambiguity is most notable in the contour landmarks in semi-frontal and profile poses, as visible in Figure 3.7. 3D landmarks maintain the physical meaning of the contour across the whole range of head poses, while 2D landmarks change semantics which introduces additional complexity in the training process.

Several different approaches have emerged, seeking to exploit the coherence of 3D facial

Table 3.5: Deep learning methods summary.

Methods	Architecture	Positions	Heatmaps	Year
CF-CNC [86]	Cascaded CNNs	Yes	No	2013
CFAN [87]	Cascaded auto-encoders	Yes	No	2014
DVLN [81]	Two coupled CNNs	Yes	No	2017
MCNet [84]	Single CNN	Yes	No	2017[a]
DCNC [61]	Cascaded local & shared CNNs	Yes	No	2017
DAN [88]	Cascaded CNNs	Yes	No	2017[b]
FEC-CNC [72]	Cascaded local CNNs	Yes	No	2017
FAN [92]	Stacked HGs	No	Yes	2017[c]
DSRN [82]	Single CNN	Yes	No	2018[d]
LAB [95]	Stacked HGs	No	Yes	2018[e]
DeCaFA [91]	Cascaded U-nets	Yes	Yes	2019
AWing [98]	Stacked HGs	No	Yes	2019[f]
Liu et al. [96]	Stacked HGs	No	Yes	2019
KDN [97]	Stacked HGs	No	Yes	2019
HRNet [99]	Parallel high-to-low CNNs	No	Yes	2020[g]

[a] `https://github.com/ZhiwenShao/MCNet`
[b] `https://github.com/MarekKowalski/DeepAlignmentNetwork`
[c] `https://www.adrianbulat.com/face-alignment`
[d] `https://github.com/xinxinmiao/DSRN`
[e] `https://wywu.github.io/projects/LAB/LAB.html`
[f] `https://github.com/protossw512/AdaptiveWingLoss`
[g] `https://github.com/HRNet/HRNet-Facial-Landmark-Detection`

structure mostly differing in the representation of the regression target. However, any 3D alignment approach needs reliable 3D annotations along with the images. One way of obtaining the necessary ground-truth information is by utilizing specialized 3D imaging hardware, which produces a 3D point cloud corresponding to the pixels in the image [100, 101, 102]. These raw results can not be used directly because each facial scan has a different topology of vertices and must be registered under a single mesh topology. This is usually done by employing an Iterative Closest Point (ICP) algorithm and its variants [103, 104]. Due to the complicated acquisition process, these data sets are collected in a controlled environment with relatively few subjects.

A different approach to building a 3D alignment data set is to fit a 3D Morphable Model [12, 105, 106] on existing large 2D data sets. 3DMM is a statistical model of the face shape built from a data set of registered facial scans. Since it is a vital part of the whole 3D face alignment pipeline, we will introduce the general concept of 3DMM construction and representation.

We can define the 3D face shape (mesh) of N vertices as a $3N \times 1$ vector of their 3D coordinates:

$$\mathbf{S}_{3D} = [x_1, y_1, z_1, \ldots, x_N, y_N, z_N]^T \tag{3.4}$$

Using Principle Component Analysis (PCA) and its variants on a data set of registered 3D shapes, a 3DMM can be constructed and defined in the following way:

$$\mathbf{S}^N_{3D} = \bar{\mathbf{S}}_{3D} + \sum_k^{N_I} p^I_k \mathbf{S}^I_k + \sum_k^{N_E} p^E_k \mathbf{S}^E_k \tag{3.5}$$

where $\bar{\mathbf{S}}_{3D}$, $\mathbf{S}^I$, and $\mathbf{S}^E$ represent the mean shape, identity or face structure bases, and expression or action bases, respectively (see Figure 3.8). The corresponding parameters which control their linear combination in the 3DMM are represented by $\mathbf{p^I} = [p^I_1, \ldots, p^I_{N_I}]$ and $\mathbf{p^E} = [p^E_1, \ldots, p^E_{N_E}]$. They are often combined in a single parameter vector $\mathbf{p} = [\mathbf{p^I}, \mathbf{p^E}]$ controlling the non-rigid transformations of the 3DMM:

$$\mathbf{S}^N_{3D} = \mathscr{N}(\mathbf{p}) \tag{3.6}$$

The resulting 3D mesh $\mathbf{S}^N_{3D}$ is in normalized shape space. In order to bring it to the 2D space on the image plane, a model of the camera with its transformations needs to be included. Weak perspective projection of the pinhole camera model is usually employed with six degrees of freedom (scale, three rotations, and two translations), which can be represented by a vector $\mathbf{c} = [s, r_x, r_y, r_z, t_x, t_y]$. The projection can then be defined as:

$$\mathbf{U} = \mathscr{W}(\mathbf{p}, \mathbf{c}) \equiv \mathscr{P}(\mathscr{N}(\mathbf{p}), \mathbf{c}) \tag{3.7}$$

where $\mathbf{U} = [x^u_1, y^u_1, \ldots, x^u_N, y^u_N]$ represents the projected 2D coordinates in the image. Finally, a subset of projected 2D coordinates $\mathbf{U_L}$ corresponding to the annotated 2D landmarks in the

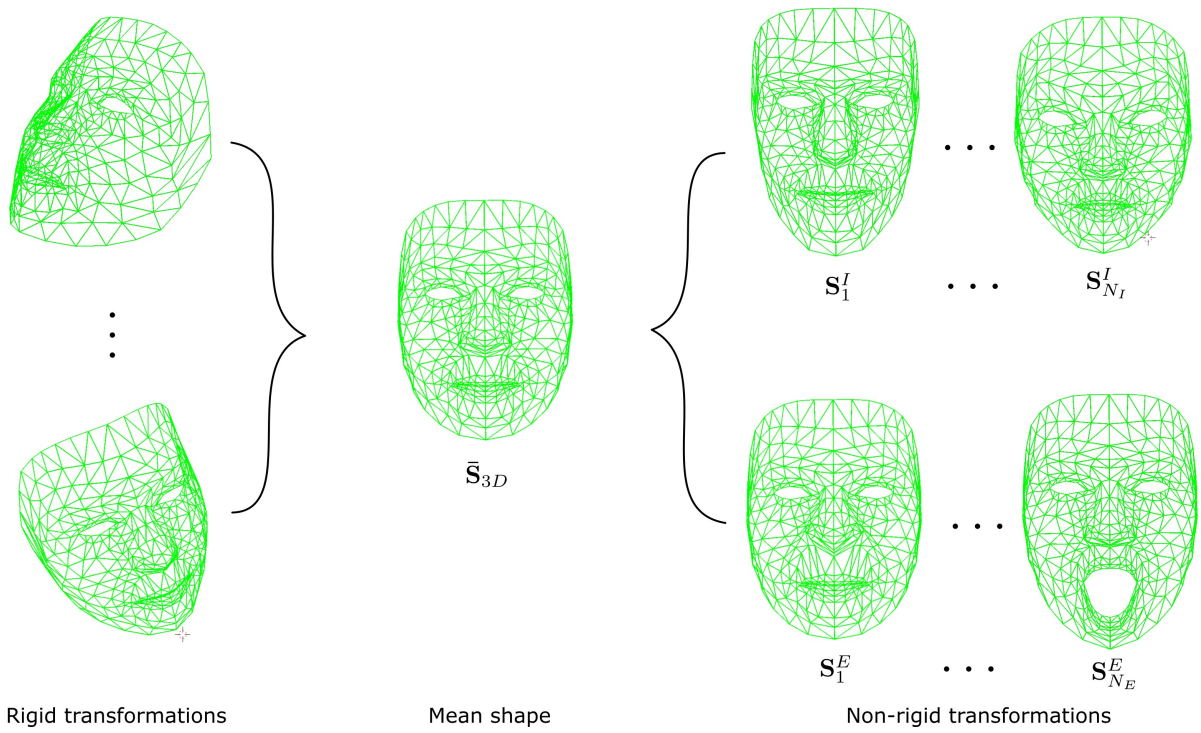


Figure 3.8: Components of the 3DMM.

image can be used to drive the optimization process to determine the "ground-truth" 3D annotations:

$$\arg\min_{(\mathbf{p},\mathbf{c})} \|\mathscr{W}_L(\mathbf{p},\mathbf{c}) - \mathbf{S}^*\|^2 \tag{3.8}$$

The goal of the optimization process is to find the rigid ($\mathbf{c}$) and non-rigid ($\mathbf{p}$) transformation parameters of the 3DMM that minimize the distance between the annotated 2D landmarks and their corresponding 3D shape projections. In order to mitigate the inconsistency of the 2D landmarks, dynamic correspondence of the contour landmarks is employed, such as landmark marching [107]. This optimization can be performed on large and diverse 2D alignment data sets to produce an "in-the-wild" 3D alignment data set under consistent annotations with minimal supervision [92].

We classified the existing approaches by the regression target and, consequently, their output into three categories. Sparse 3D alignment methods optimize the subset of 3D mesh vertices or their projections corresponding to the usual 2D mark-up. 3DMM alignment methods optimize the 3DMM transformation parameters $\mathbf{p}$ and $\mathbf{c}$ to produce a dense 3D shape by applying the transformations. Finally, direct 3D alignment methods do not utilize a 3DMM but can still produce a dense 3D shape through creative output and optimization design. Each group of methods will be covered more thoroughly in the following sections.

3.2.1 Sparse 3D alignment

Since 2D alignment was extensively researched in the last 20 years, the straightforward approach is to simply replace the 2D annotations with the corresponding 3D annotations or their projections [108, 109, 110]. As already mentioned in section 3.1.5, Bulat and Tzimiropoulos experimented with both 2D and 3D alignment using the same CNN architecture (named 2D-FAN and 3D-FAN) where they observed a slight improvement in accuracy by switching to 3D alignment [92]. The model 3D-FAN was trained on the 300W-LP-3D [111] data set constructed by 3DMM fitting on the 300W data set.

In recent work, Deng et al. use both 2D annotations and the corresponding 3D projections to drive the optimization process in a cascaded framework [112]. Hourglass CNN architectures are used in two stages. The first stage is used for coarse alignment and joint estimation of 2D landmarks in both frontal and profile poses, exploiting the correspondence of different annotations. The second stage optimizes the corresponding projected sparse 3D landmarks to refine the prediction and provide full-pose alignment results.

One of the earliest attempts at 3D alignment was proposed by Jourabloo and Liu in their work called Pose-Invariant Face Alignment (PIFA) [113]. As the name suggests, they were motivated to achieve face alignment covering full profile poses even. This method is interesting because it can be seen as a predecessor for 3DMM alignment methods. A model is constructed from 3D facial scans and used to produce 3D annotations on existing 2D alignment data sets. However, the 3D Point Distribution Model (3DPDM) is constructed using a sparse set of landmarks and can not produce a dense 3D mesh. A standard cascaded regression approach is used where optimization alternates at each stage between rigid and non-rigid transformation parameters.

3.2.2 3DMM alignment

In later work, the same authors extended their approach by utilizing a 3DMM to produce a dense 3D output and replaced hand-crafted features with CNNs [114]. With an additional introduction of 3D-aware inputs for CNN, they managed to improve their previous results. A similar approach was presented in [111] by Zhu et al. named 3D Dense Face Alignment (3DDFA). The cascaded regression approach was used to iteratively update the 3DMM transformation parameters with CNNs as stage regressors. In addition to the standard RGB input to the CNN, a novel channel called Projected Normalized Coordinate Code (PNCC), specifically designed to transfer the output of previous stages of the cascade, was used as input as well. A very important additional contribution is a method to expand the 300W training set with 3D annotations and augmenting it with generated profile samples to create the 300W-LP data set.

In an attempt to connect the stages of the cascaded regression framework in an end-to-end

manner, a technique already proven beneficial for 2D alignment, Jourabloo et al. introduced a visualization layer between the stages in their recent work [115]. The differentiable visualization layer generates an image of a 3D face using surface normals based on currently estimated 3DMM parameters to be used as an additional input to the next CNN stage. It allowed the flow of information between the deep learning stages and an end-to-end optimization leading to faster training convergence.

Liu et al. focused on the supervision signal and the loss function of the 3D alignment training in their work Dense Face Alignment (DeFA) [116]. In addition to the standard sparse supervision (distance between 2D annotations and projections of the corresponding vertices), two additional terms to the loss function were added. The first term includes SIFT matching of vertices on pairs of images of the same face enabling dense alignment. The other term includes visible contour supervision using Holistically-nested Edge Detection (HED) [117] as the ground-truth. Additional supervision signals allowed them to achieve high accuracy using a single CNN without cascading.

Finally, Bhagavatula et al. emphasized the limitations of the 3DMM and its flexibility to model unseen faces [118]. Their approach is based on a 3D Spatial Transformer Network (3DSTN), which estimates the camera projection matrix as usual and parameters of the Thin Plate Spline (TPS) [119] warping function, which performs the non-rigid 3D shape transformation. Using a non-linear warping function eliminated the indirect need for large data sets of 3D facial scans required by the 3DMM. This method shares the same goal as the next group of methods and can serve as a transitional example.

3.2.3 Direct 3D alignment

The latest direction for 3D face alignment is to skip the 3DMM and its constraints and directly regress a dense 3D face shape. The straightforward approach of simply using a fully connected layer with an output for each vertex coordinate is not feasible due to the large number of vertices. Such a layer would be impractically large and challenging to train.

Nevertheless, Jackson et al. recently proposed the first direct 3D alignment method called Volumetric Regression Network (VRN), which uses a volumetric representation of the 3D face shape [120]. Such a representation allows them to use a fully convolutional network architecture and convert the problem into a 3D binary volume segmentation. The 3D face shape is discretized into voxels, a 3D binary volume, where the label of the voxel represents if it belongs to the face or the background. Two stacked hourglass CNNs are used with RGB images as input and binary volume as output. Another example of a direct representation was presented by Yu et al. using a per-pixel 2D flow between the input image and the synthetically rendered image of a 3DMM [121]. An encoder-decoder architecture was trained on both synthetic and real examples (300-W-LP).

In later work, a more efficient direct representation was proposed by Feng et al. named Position Map Regression Network (PRNet) [122]. A UV position map is used wherein the RGB values for each UV texture coordinate of the 3D face model are replaced by the 3D coordinates of the vertices allowing a fully convolutional architecture. However, it is more efficient than the volumetric representation [120] which needs to discretize the head interior as well, which is redundant for the face alignment problem. Such an approach allowed them to use a single lightweight CNN architecture and achieve superior accuracy.

3D face alignment is a necessary step in the right direction if we want to achieve robustness across the full range of head poses since manual annotation of self-occluded landmarks is not feasible. The sparse 3D alignment methods benefit directly from the consistent and complete annotations even in full profile poses. In order to efficiently achieve dense alignment producing a detailed 3D facial mesh, 3DMM alignment methods optimize the 3DMM rigid and non-rigid parameters. Finally, direct 3D alignment methods eliminate the constraints of the 3DMM and directly optimize the dense 3D shape representation. All of these methods achieve full pose face alignment that 2D alignment methods can not achieve by design.

The biggest obstacle, however, for wider adoption of this approach is the lack of annotated large-scale 3D data sets. The 3DMM is constructed from a data set of 3D facial scans using PCA, which means that the model flexibility directly depends on the sample variance. The current publicly available data sets with 3D facial scans are collected on a scale of a hundred subjects. Collecting such a data set is cumbersome and expensive because of the additional hardware requirements. On the other hand, 2D alignment data sets contain "in-the-wild" images of thousands of subjects. Automatic re-annotation of these data sets using 3DMM fitting is a viable alternative. However, even though direct 3D alignment methods eliminate the explicit constraint of the 3DMM, it is still there implicitly through the construction of the training sets.

Nevertheless, recent years have seen an advancement of depth cameras, making them smaller and cheaper to the point of integrating such cameras into mobile phones. With such advancements, the barriers for large-scale data set collection are becoming smaller, making this approach viable in the future. The summary of the described methods is presented in Table 3.6.

3.3 Multi-task learning

Multi-task learning has proven to be effective in many research areas [123]. One of the first attempts for face alignment was proposed by Zhang et al. in their work named Tasks-Constrained Deep Convolutional Network (TCDCN) [124]. The main idea is to jointly train auxiliary attributes with landmark detection (Figure 3.9). They proved that the complexity of the shape detection problem could be reduced by learning auxiliary relevant attributes. However, modi-

Table 3.6: 3D face alignment summary.

Methods	Architecture	Target	Year
Tulyakov & Sebe [108]	Cascaded regression	Sparse 3D shape	2015
Gou et al. [109]	Cascaded regression	Sparse 2D shape + 3DMM	2015
PIFA [113]	Cascaded regression	3DPDM	2015
Zhao et al. [110]	Single CNN	Sparse 3D shape	2016
3DDFA [111]	Single CNN	3DMM	2016[a]
3D-FAN [92]	Stacked HGs	Sparse 3D shape	2017[b]
PAWF [114]	Deep cascaded regression	3DMM	2017
Jourabloo et al. [115]	Deep cascaded regression	3DMM	2017
DeFA [116]	Single CNN	3DMM	2017[c]
3DSTN [118]	Single CNN	3DMM + TPS	2017
VRN [120]	Stacked HGs	3D volume	2017[d]
Yu et al. [121]	Encoder-decoder	2D flow	2017
CMHM [112]	Cascaded HGs	Sparse 2D/3D shape	2018
PRNet [122]	Encoder-decoder	UV position map	2018[e]

[a] `https://github.com/cleardusk/3DDFA`
[b] `https://www.adrianbulat.com/face-alignment`
[c] `http://cvlab.cse.msu.edu/project-pifa.html`
[d] `http://aaronsplace.co.uk/papers/jackson2017recon/`
[e] `https://github.com/YadiraF/PRNet`

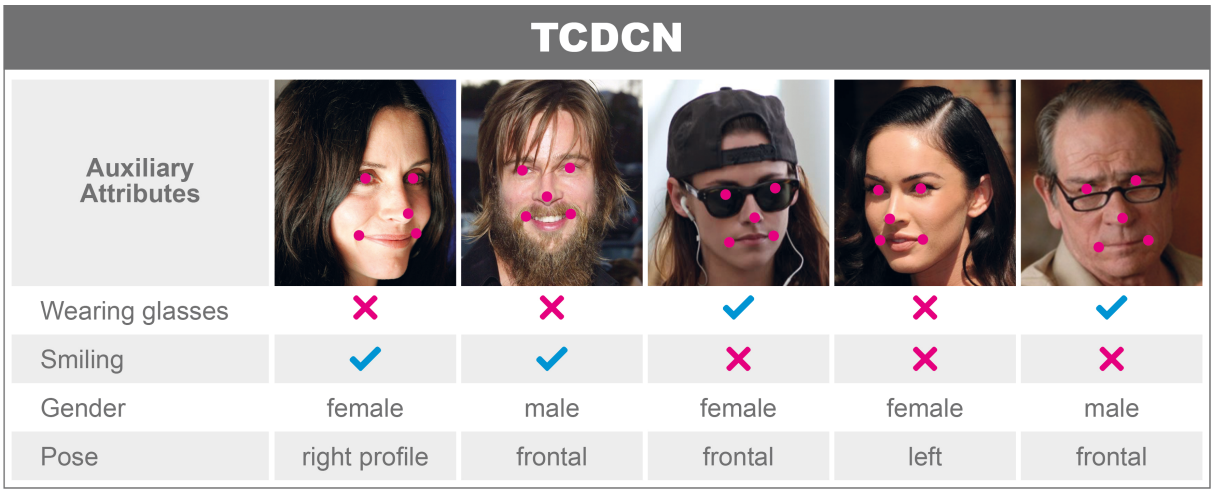


Figure 3.9: Some of the attributes used in TCDCN [124].

fications to simple multi-task learning algorithms were needed because of different task complexities and convergence rates. The authors introduced inter-task correlation modeling to the objective function via the covariance matrix to improve the performance and analyze relations between attributes and landmarks. Additionally, a dynamic task coefficient was introduced to address the problem of different convergence rates between tasks. Thus, the learning process of some tasks could be turned off, or the impact on the objective function reduced if needed. Twenty-two different attributes were used and annotated in the training set.

Another early attempt was presented by Zhao et al. by modeling and exploiting relationships between multiple face analysis tasks (head pose, facial expression, and landmark detection) for mutual benefit. This unified method is called iterative Multi-Output Random Forests (iMORF) [125]. Random patches of the face image are used similarly to CRF [126], and a hybrid cost function is optimized, which models the quality of each task using associated weights. The weights are dynamically adapted to emphasize the head pose at the top nodes until sufficient classification purity is achieved. Afterward, facial expressions take precedence, again, until sufficient purity is reached. Lastly, the landmark regression is performed. In the next phase, cascaded regression is employed to refine the face analysis estimations further. In addition to the shape-indexed appearance features, shape-related features are added that are modeled as distances and ratios of the landmark positions.

Face alignment or facial landmark detection highly depends on face detection, making it logical to combine these two problems under a multi-task learning framework. Chen et al. in [127] demonstrated that alignment helps detection and managed to obtain improved results using joint learning in a standard cascaded regression framework using boosted regression trees as in [45].

Later on, the same idea was examined in [128] using CNNs. Again, a cascaded regression framework is used with three stages of CNNs where each stage has a different goal under a paradigm of coarse-to-fine refinement. These tasks are performed sequentially in the cascade: face region proposal, face bounding box refinement, and face alignment with five landmarks. Additionally, each stage can reject the region as a non-face, meaning it simultaneously performs face classification. This work was subsequently extended by an additional stage using a Multi-view Hourglass Model (MHM) [129] to produce a dense set of facial landmarks exploiting the correspondence of semi-frontal and profile annotations of the Menpo data set.

An interesting combination of face alignment and segmentation was proposed by Zhao et al. in their multi-task learning work [130]. An encoder-decoder CNN architecture is used where the encoder is conditioned for the face alignment task, and the decoder estimates the segmentation mask. A boost in accuracy is achieved for both of these highly correlated tasks through joint optimization.

It is evident from described methods that face alignment can benefit from multi-task learning

Table 3.7: Multi-task learning methods summary.

Methods	Architecture	Additional tasks	Year
TCDCN [124]	Single CNN	Auxiliary attributes	2014[a]
iMORF [125]	Cascaded regression	Head pose + facial expression	2014
Chen et al. [127]	Cascaded regression	Face detection	2014
Zhang et al. [128]	Deep cascaded regression	Face detection	2016
MHM [129]	Cascaded HGs	Face detection	2019
Zhao et al. [130]	Encoder-decoder	Face segmentation	2019

[a] `http://mmlab.ie.cuhk.edu.hk/projects/TCDCN.html`

with related tasks such as face detection, head pose estimation, expression classification, gender estimation, etc. This is especially useful for deep learning and cascaded regression frameworks where more general facial features can be learned and shared across related problems. Additionally, knowledge between tasks is often complementary and can boost accuracy (e.g., smiling expression and landmark detection surrounding mouth region). The summary of the described methods is presented in Table 3.7.

3.4 Occlusion modeling

Faces are often occluded in unconstrained scenarios, which represents a challenging obstacle for accurate face alignment. Different occlusion sources are frequently seen covering the face, such as accessories (e.g., hats and glasses), beards, different hairstyles, and self-occlusion due to extreme head poses. Despite this, humans can quite accurately estimate a person's face shape, while machine learning models often fail and produce unstable estimates. It is an important problem to address and has thus attracted the attention of the research community.

One of the first methods that explicitly handles occlusions was presented in [131] and was named Robust Cascaded Regression (RCPR). Burgoss-Artizzu et al. extended CPR [53] and ESR [45] methods and introduced a new, more challenging data set called Caltech Occluded Faces in the Wild (COFW), which has become a benchmark data set for face alignment with occlusions. It is publicly available and has annotations for occluded landmarks. At the beginning of the RCPR approach, the face image is divided into nine regions. At each stage, multiple regressors (as in [45]) are trained, where each regressor is allowed to extract features from only one of the nine regions. Each such two-level boosted regressor learns to predict the occlusion of the corresponding region along with the landmark positions as the third dimension of the output vector. The predicted occlusion level from the previous stage is then used to assign the

weights when combining current estimates of different regressors. It was the first method that also predicted occlusion with face alignment.

A different occlusion handling approach was taken in [132]. The method was named Regional Predictive Power (RPP), with the main idea to apply a graph-based segmentation of face images. The usefulness of each segment for the face alignment task is determined by using sieving regression forest votes [50]. A confidence value (if it is a face region) is obtained for each pixel, based on the center sieve from random forest votes. The RPP map is used to assign the weights to regressors from either of these methods: CPR [53], RCPR [131], ESR [45]. The weights were calculated by accumulating confidences from all the pixels used in those regressors.

Wu et al. use cascaded regression with explicit occlusion learning [133]. At each stage, landmark visibility probabilities are estimated first, then the landmark locations. The visibility probability updates at each stage are trained using extracted SIFT features around the current landmark positions concatenated with the shape features. The shape features are formed using the difference between pairwise landmark locations. Additionally, an occlusion pattern loss term is added to the standard least-squares objective function, which penalizes improbable occlusion patterns. The loss function is constructed from auto-encoder network reconstruction errors. The landmark localization is trained in the same way and with the same features. However, the appearance part of the features is weighted by the visibility probabilities. Missing annotation is handled by adding binary weights to the weighted least squares problem.

While all of the above methods try to estimate occluded facial regions to avoid feature extraction from those regions, Zhang et al. propose a different approach in their work called Deep Regression Networks Coupled with De-corrupt AutoEncoders (DRDA) [134]. The aim is to reconstruct the occluded region using de-corrupt auto-encoders, again, in a cascaded regression framework. Deep regression and de-corrupt auto-encoder alternate each stage in order to benefit from each other. Estimated landmark positions are used to partition the face. The cropped image is fed into the auto-encoder to produce the un-occluded version, which is then forwarded as input to the next deep regression stage. The method produces realistic images without occlusions which in turn improves alignment accuracy.

It is expected from face alignment methods to show a certain level of robustness to occlusions to be usable in real-life situations. The main approach in the literature is to estimate the level and region of occlusion in the image of the face to avoid extracting ambiguous features which cause unstable face shape predictions. However, with the recent introduction of generative models such as auto-encoders or Generative Adversarial Networks (GANs) [31], a new approach has emerged where the occluded region is reconstructed and used for accurate face alignment. An obvious drawback of this new direction is the increase in computational complexity. However, with optimized computing on GPUs, these methods demonstrate the potential

Table 3.8: Occlusion modeling methods summary.

Methods	Architecture	Occlusion modeling	Year
RCPR [131]	Cascaded regression	Hand-crafted regions	2013
RPP [132]	Cascaded regression	Segmentation	2015
Wu et al. [133]	Cascaded regression	Landmark visibilities	2015
DRDA [134]	Deep cascaded regression	Decorrupt auto-encoder	2016

to surpass human accuracy for occluded face alignment. A summary of the described methods is presented in Table 3.8.

3.5 Facial expression recognition

As already mentioned in the introduction, Ekman and Friesen discovered six invariant, prototypic emotions (anger, disgust, fear, happiness, sadness, and surprise) ideal for automatic recognition. One important drawback of this model became evident, though. It is too crude to accurately model the complexity of emotions people experience in everyday lives. As a response, Facial Action Coding System (FACS) [135] was developed to define atomic facial muscle movements named Action Units (AU) spanning the whole spectrum of human facial expressions. Its aim is objectivity in the signal measurement, which is separated from the final expression classification, often influenced by the context. Consequentially, a group of researchers [136, 137, 138, 139, 140] tried to develop algorithms that recognize these simpler, intermediate categories and synthesize the final expression afterward. On the other hand, FACS annotation is a very tedious process that requires expert knowledge few people possess. Therefore, few data sets with full FACS annotations are available to the community making this approach less researched.

FER is traditionally comprised of three distinct steps: face detection, feature extraction, and classification. In most papers, face detection is not discussed in detail since the face location and size are assumed as *a priori* knowledge. The greatest emphasis is put on feature selection and extraction, which is often considered to be the critical part of the system. On the other hand, standard machine learning techniques are mostly used for the classification step. The used features can roughly be divided into appearance and geometric-based. The appearance features are extracted from facial image intensities to represent a discriminative textural pattern, while the geometric ones need accurate landmark positions to construct different relations. The geometric features are very sensitive to the individual face shape configuration and are therefore

less consistent in person-independent scenarios. It is important to note that these two types of features have recently been shown to be complementary [141]. Hence, hybrid systems similar to the one proposed in this thesis are gaining popularity.

An additional direction of research is to integrate temporal dimension into both appearance and geometric features when working with image sequences [142, 143, 144, 145, 146]. However, this thesis focuses on single static image recognition since it is a natural first step that can be extended in future work.

3.5.1 Hand-crafted features

Well known and widely successful hand-crafted features such as variations of Local Binary Patterns (LBP) [7, 8, 142, 143, 147, 148, 149, 150, 151, 152, 153] and Histogram of Oriented Gradients (HoG) [148, 149, 152, 154], Gabor filters [147, 151, 155, 156, 157, 158] and Local Phase Quantization (LPQ) [152, 154] descriptors have also been considered for FER. While most approaches considered a regular grid of patches [142, 143, 147, 148, 149, 150, 151, 159, 160] or the whole face region [154, 156, 158] for feature extraction, there have been advances in determining common and specific salient facial regions for each expression. In [7], Happy and Routray demonstrated the importance of facial landmark detection to find the salient patches from which they extract features. Through the use of a one-vs-one SVM classifier for each patch and each expression pair, they were able to find the most discriminative patches for each expression. A similar idea was adopted in [8]. However, a regular grid of patches was used without landmark detection, which resulted in lower accuracy than in [7]. In [153], Khan et al. performed a psycho-visual experiment to track the participant's gaze and determine which regions of the face are salient for specific expression. Rivera et al. designed a novel descriptor called Local Directional Number Pattern to differentiate between bright and dark transitions, which occur often in faces [159].

3.5.2 Feature fusion

On the other hand, some researchers [148, 151, 152, 154] tried to fuse different texture encoding features to extract complementary information that would benefit the FER. For instance, Zhang et al. used multiple kernel learning to combine two different feature representations: HoG and LBP [148]. A different approach to feature fusion was taken in [151] where a pool of SVM classifiers was trained using either Gabor filters or LBPs as features. A genetic algorithm was then used to find the optimal ensemble of classifiers in terms of both size and accuracy. The fusion idea was tested with geometric features as well [155, 161]. Wan et al. used the Constrained Local Model (CLM) to detect the facial landmarks and used their positions normalized to the mean shape as geometrical features, which they concatenated to Gabor features

as input to Robust Metric Learning [155]. The method was developed to recognize spontaneous expressions.

3.5.3 Deep learning

While all of the previously mentioned methods use hand-crafted and heuristically determined features, experiments with deep learning using CNNs [35] on the FER problem were recently conducted as well [162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172]. As already mentioned in the introduction, deep learning methods have serious over-fitting problems with small datasets that are typical for FER. Several different approaches have recently been examined in order to cope with the mentioned problem: artificial data augmentation, data set merging, and transfer learning. For a more in-depth review of FER methods using CNNs, we refer the reader to a recent survey by Pramerdorfer and Kampel [172]. Additionally, they demonstrate that modern architectural changes in deep networks reduce the over-fitting problem on a moderately large FER 2013 data set (35k images) [173].

Kim et al. used a combination of both aligned and non-aligned faces to train their ensemble of deep CNNs (DCNNs), making the method more robust to face registration problems on faces in the wild [170]. Levi et al. also used an ensemble of twenty DCNNs, each having a differently preprocessed input [167]. They designed a novel transformation of image intensities into 3D space called mapped LBP to reduce the illumination variation in the training set. The mapped LBP transformations with different parameters were used as one of the inputs in the ensemble, along with ordinary RGB intensities. Lopes et al. tried standard preprocessing techniques (image normalizations, synthetic samples, etc.) and achieved state-of-the-art results on the CK+ benchmark dataset [163]. In [164], the authors combined seven different data sets to have enough samples for each expression to train on, making it hard to compare to other methods, which restricted their training samples to those available in the individual benchmark data sets.

Finally, transfer learning has recently emerged as the most effective approach to small data set sizes [174, 175]. Ng et al. used a general object recognition pre-trained DCNN model and fine-tuned it in two stages. In the first stage, they used the large FER 2013 data set and the SFEW 2.0 training set in the second. However, both Levi et al. and Zhai et al. achieved better results using a model pre-trained on a related face recognition task with extremely large data sets (millions of images) [167, 169]. State-of-the-art results on the SFEW 2.0 data set were achieved by Yu et al. using an ensemble of DCNNs, data augmentation (random affine transformations), and pre-training on the larger FER 2013 data set. An interesting approach to transfer learning was presented in [168]. The authors trained a DCNN for FER using a face recognition model's convolutional weights as regularization. Next, they appended fully connected layers and fine-tuned the network for a specific data set. Since they used a single DCNN, the authors achieved an impressive run-time speed (3 ms). However, they require a high-end GPU (TitanX), which

is not viable for mobile and embedded platforms.

Even though deep learning methods achieve good results, problems with over-fitting and slow run-time remain, confirming the need for an effective and fast FER method.

Chapter 4

Globally initialized facial landmark detection using local binary features

The proposed method aims to efficiently estimate 2D landmark positions from facial images with known face location and size. As explained in section 3.2, 3D alignment methods achieve higher robustness to extreme head poses but lack the fine precision of 2D alignment methods, which is more important for face analysis problems. The optimal balance between high efficiency and accuracy is demonstrated by cascaded regression architectures (section 3.1.2) with two key components as discussed in the following paragraphs.

Deep global initialization As can be seen from deep learning methods [72, 88, 124] in section 3.1.5, CNNs are well suited for coarse face alignment (high accuracy on the challenging 300-W subset) due to a couple of reasons. Firstly, CNNs use global features taking into account the whole face holistically and the context as well. It makes it easier to infer the global orientation of the face and head. Secondly, CNNs possess high capacity and flexibility to absorb extreme appearance variations due to different head poses and backgrounds.

Shape-indexed local features Gradual alignment helps to improve both accuracy and robustness, as can be seen from the large number of methods adopting cascaded regression in section 3.1.2. However, local shape-indexed features in later stages of the cascade seem to be important as well, providing the algorithm an attention mechanism necessary for fine-grained alignment. For instance, Kowalski et al. used global CNNs in all stages of the cascade. However, in order to achieve competitive results, a heatmap constructed from current landmark positions needed to be passed to the next stage of the cascade. It helped later stages focus on relevant regions of the face. Even so, the method achieved significantly worse results on the common 300-W subset than methods utilizing local shape-indexed features (DCFE [63], FEC-CNC [72], DCR [66]) while maintaining similar results on the challenging subset.

The proposed implementation of the presented concepts will be described with more details in the following two sections 4.1 and 4.2. The evaluation on the benchmark data sets with run-time comparisons will be given in section 4.3 and result discussion presented in section 4.4.

4.1 Global initialization

The most important aspect of this component is the selection of the CNN architecture. Most deep learning methods in section 3.1.5 use large networks based on well-known architectures: AlexNet [72], VGGNets [81, 84, 88], and ResNets [89, 92]. These architectures were developed for general object detection and classification, which is usually the driver in CNN architecture design as explained in section 2.2.2. This problem, however, is more complex with numerous different classes of objects, each with variations of its own represented in millions of images. It naturally leads to deeper and larger designs that are over-capacitated for face alignment leaving room for optimization.

Large efforts have recently been invested by the research community in neural network architecture optimization to produce realistic models of practical size and performance. This resulted with a series of innovations in the form of new types of convolution layers (depthwise separable and pointwise [176], dilated [177], mixed [178] convolutions), blocks of layers (bottleneck residual [179], squeeze and excite [180], shuffle [181], ghost [182], fire [183] blocks), and automated neural architecture search [184, 185, 186, 187, 188]. These architectures are, however, optimized for a more complex problem resulting in complex designs often poorly supported on different platforms. The performance improvements are measured in the theoretical number of floating-point operations (FLOPs), which doesn't necessarily translate to faster execution time since memory access (often the real bottleneck) is not accounted for.

Our aim, then, is to use standard and well-supported deep learning layers to design an efficient CNN architecture for the initial global stage in the cascaded regression framework for face alignment. Inspired by the simple and lightweight tiny version of the YOLO9000 object detection method [189], we modified the backbone architecture for this purpose. A neural network can be optimized in the following ways: the number of filters in each convolution layer (network width), the total number of layers (network depth), and input resolution. Unlike general object detection, we are interested in a single object, namely, the face. There is no need for a high-resolution input, especially for the initial coarse stage. Additionally, the final layers are adjusted to reduce the number of weights and produce 2D landmark coordinates. These adjustments resulted in an architecture presented in Table 4.1. The input is a loose, gray-scale crop of the face bounding box (allowing for enough context) resized to a 96×96 resolution. The network output is a vector of 2D landmark coordinates of length $2 \times L$ where L represents the number of landmarks. The presented architecture has approximately one million trainable parameters

Table 4.1: CNN architecture used for the global initial stage.

Layer type	Filters	Size/Stride	Output
Convolutional	16	$3 \times 3/1$	$96 \times 96 \times 16$
Maxpool		$2 \times 2/2$	$48 \times 48 \times 16$
Convolutional	32	$3 \times 3/1$	$48 \times 48 \times 32$
Maxpool		$2 \times 2/2$	$24 \times 24 \times 32$
Convolutional	16	$1 \times 1/1$	$24 \times 24 \times 16$
Convolutional	128	$3 \times 3/1$	$24 \times 24 \times 128$
Convolutional	16	$1 \times 1/1$	$24 \times 24 \times 16$
Convolutional	128	$3 \times 3/1$	$24 \times 24 \times 128$
Maxpool		$2 \times 2/2$	$12 \times 12 \times 128$
Convolutional	32	$1 \times 1/1$	$12 \times 12 \times 32$
Convolutional	256	$3 \times 3/1$	$12 \times 12 \times 256$
Convolutional	32	$1 \times 1/1$	$12 \times 12 \times 32$
Convolutional	256	$3 \times 3/1$	$12 \times 12 \times 256$
Maxpool		$2 \times 2/2$	$6 \times 6 \times 256$
Convolutional	64	$1 \times 1/1$	$6 \times 6 \times 64$
Convolutional	512	$3 \times 3/1$	$6 \times 6 \times 512$
Convolutional	64	$1 \times 1/1$	$6 \times 6 \times 64$
Convolutional	512	$3 \times 3/1$	$6 \times 6 \times 512$
Convolutional	128	$1 \times 1/1$	$6 \times 6 \times 128$
Avgpool		$2 \times 2/2$	$3 \times 3 \times 128$
Linear		Global	$2 \times L$

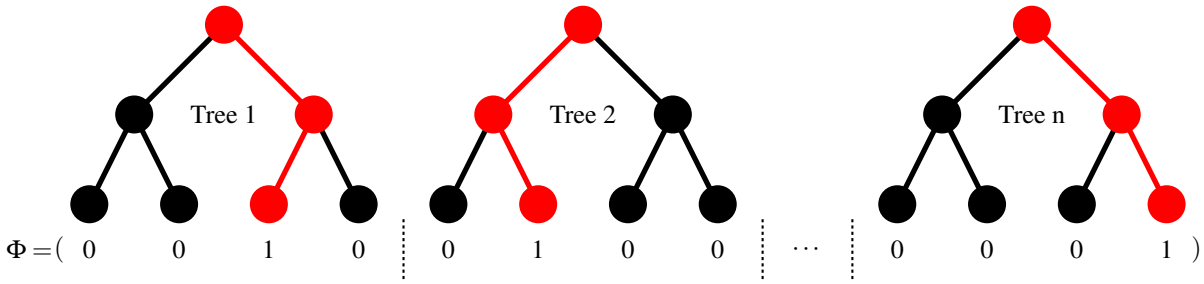


Figure 4.1: The process of generating a sparse feature vector Φ with an ensemble of n decision trees. The path that a sample takes through each tree is drawn in red. Exactly n components of Φ are set to 1 based on these paths. The rest are set to 0.

resulting in a small and computationally efficient model as demonstrated in section 4.3.3.

4.2 Local shape-indexed features

As already highlighted, local shape-indexed features are excellent for fine-grained alignment in deeper stages of the cascaded regression framework. Limiting the feature space using local patches centered on coarsely aligned landmarks allows for more efficient and relevant feature extraction. We propose to achieve this using Pixel Difference Features and ensembles of decision trees.

Decision trees [14] are a tried-and-true machine learning method with a long tradition. They are especially powerful when combined in an ensemble [16] (the outputs of multiple trees are usually summed together). A nice property of decision-tree ensembles is that the method easily deals with multidimensional prediction (e.g., in multi-class classification). This is achieved by placing a vector in the leaf node of each tree. This means that the multidimensional output $\Delta(x)$ for the input sample x is computed as $\Delta(x) = \sum_i \omega_i$, where the ith vector is output by the ith tree: $\omega_i = \text{Tree}_i(x)$. As already shown in section 2.1.1, Ren et al. [18, 190] interpret this computation as a linear projection step:

$$\Delta(x) = \sum_i \text{Tree}_i(x) = \Omega\,\Phi(x), \tag{4.1}$$

where Ω is a large matrix that contains as columns the leaf-node vectors of the trees and $\Phi(x)$ is a sparse vector that indicates which columns of Ω should be summed together in order to obtain the prediction for the sample x. See Figure 4.1 for an illustration that shows how to obtain $\Phi(x)$. This interpretation enabled Ren et al. to learn an efficient method for face alignment by jointly refining the outputs of multiple decision trees [190].

If the ensemble has n trees of depth equal to d and the dimension of the output is o, then the matrix Ω has $n \cdot 2^d \cdot o$ parameters. This number can be quite large in a practical setting. In [58], we showed how to replace Equation 4.1 with a more memory-friendly computation: first, we investigated two different methods with a reduced number of coefficients; second, we

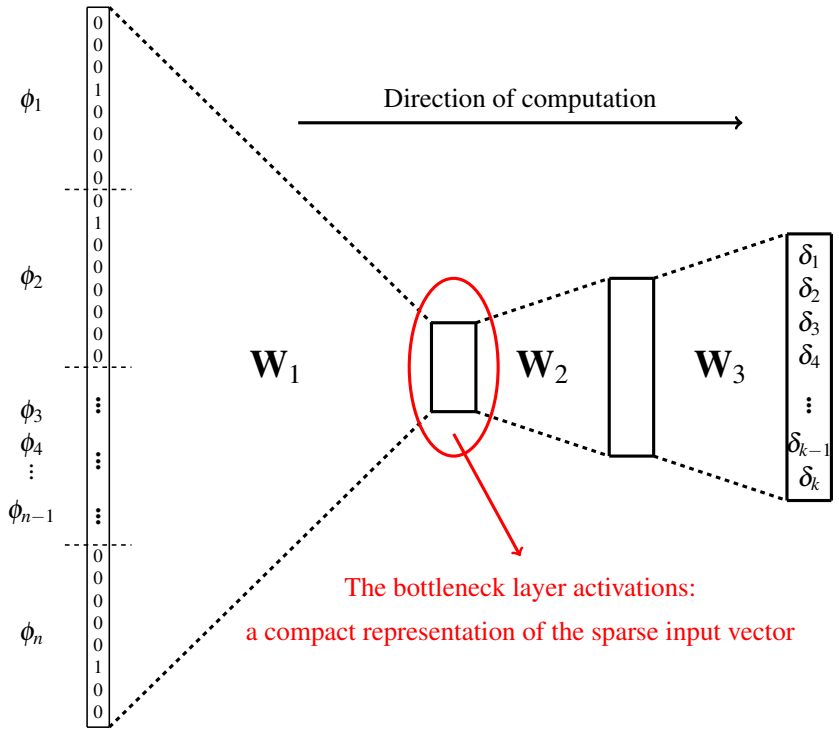


Figure 4.2: An illustration of the neural-net structure used in the experiments. First, the input sparse vector is compressed into a compact representation with the projection matrix Ω_1. Next, this representation is gradually expanded with matrices Ω_2 and Ω_3 to obtain the result. Hyperbolic-tangent nonlinearities (`tanh`) are applied in the two inner layers.

showed that the remaining coefficients can be further compressed with quantization. These ideas were applied to face alignment, which improved on the previous work of Ren et al. [190] by significantly reducing the memory requirements with no loss in accuracy.

A way to compress $\Omega \in \mathbb{R}^{o\times(n\cdot 2^d)}$ is to express it as a product of $\Omega_2 \in \mathbb{R}^{o\times r}$ and $\Omega_1 \in \mathbb{R}^{r\times(n\cdot 2^d)}$: $\Omega = \Omega_2 \cdot \Omega_1$. Of course, r has to be smaller than o. For classification, we can attempt to learn these matrices using gradient descent. For regression, besides gradient descent, we can also use the reduced-rank regression (RRR) framework [191].

Another possible path to improving memory issues is to replace linear regression for computing Δ from Φ (Equation 4.1) with a neural network (NN). One architecture that we found to work well in our experiments is

$$\Delta(x) = \Omega_3 \cdot \texttt{tanh}\left(\Omega_2 \cdot \texttt{tanh}\left(\Omega_1 \cdot \Phi(x)\right)\right), \tag{4.2}$$

where $\Omega_3 \in \mathbb{R}^{o\times 2r}$, $\Omega_2 \in \mathbb{R}^{2r\times r}$, $\Omega_1 \in \mathbb{R}^{r\times(n\cdot 2^d)}$ and `tanh` is the elementwise hyperbolic-tangent nonlinearity. See Figure 4.2 for an illustration. The matrices Ω_1, Ω_2 and Ω_3 can be trained with gradient descent through the use of backpropagation. The presented NN architecture improves memory issues only if r can be made reasonably small.

In [58], we experimentally verified that r can be significantly smaller than o when applied to face alignment, which, consequently, leads to a large reduction in storage-related issues for com-

puting Δ from Φ. More specifically, such an architecture reduces storage/bandwidth requirements by approximately three times. Further reduction is achieved by employing non-linear quantization of the weights in the long and flat Ω_1 matrix, resulting in a total improvement by a factor of twenty-one. For more details, we refer the reader to [58]. We use the NN-based method in all stages of the cascaded regression due to its scalability and simplicity, except the initial global stage.

4.3 Evaluation

This section compares the proposed novel method with recent work in this field using the 300-W benchmark data set. The results from related work will be presented as reported in their respective papers. During the years of research, several different metrics have emerged to measure the alignment accuracy. The earliest and most frequently used is the normalized mean error (NME):

$$\frac{||S - S^*||_2}{D}, \tag{4.3}$$

where D represents the normalization factor which varies between the following values in previous work:

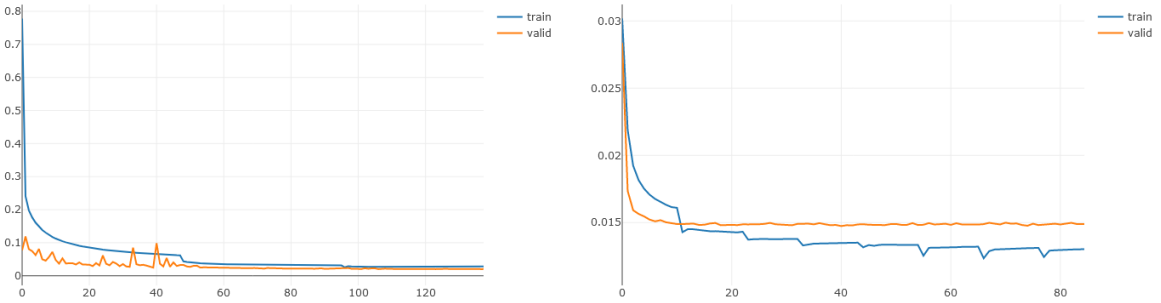
- Inter-pupil distance (IPD) - this metric is the most common one. However, it can only be used on frontal or semi-frontal faces where pupils are visible [15].
- Inter-ocular distance (IOD) - this metric is used when there are no pupil annotations and represents the distance between the outer corners of the eyes [192].
- Bounding box diagonal (BBD) - this metric is more suitable for profile faces where the first two metrics produce unreasonably small values [73].

The authors in [73] argue that these average measures are not always informative enough since few outliers can affect the result significantly and propose Cumulative Error Distribution (CED) curves which provide a much more detailed source of information for analysis. Additional measurements from the curve are then derived:

- Area Under the Curve (AUC) - this metric represents the calculated area under the CED curve up to a defined error threshold (e.g., 5% marked with $AUC_{0.05}$).
- Failure Rate (FR) - this metric represents a percentage of samples with an error greater than a set threshold (e.g., 5% marked with $FR_{0.05}$).

All of these metrics will be taken into account in addition to the inference time of the compared methods, if available.

300-W data set is currently adopted as the main benchmark data set for face alignment. It is a compilation of different data sets (AFW, LFPW, HELEN, and XM2VTS) under consistent 68-point annotation [192] with an addition of a challenging set of 135 images called IBUG. A standard partitioning was set in [15] into a training set (the training set from LFPW and



(a) (b)

Figure 4.3: (a) Training and validation errors during optimization of the initial global stage. (b) Training and validation errors during optimization of the first local stage.

HELEN, whole AFW) with 3148 images and three testing sets: the test sets from LFPW and HELEN as the common subset, the whole IBUG as the challenging subset, and both common and challenging as the full test set with 689 images. The private test set was released after the second 300 face in-the-wild challenge [193] containing 600 images.

4.3.1 Training implementation

The initial global stage was trained from scratch using the 300-W training set. We used a random 5% partition as the validation set for model selection during the training process. The training images were cropped using the annotated bounding boxes enlarged by 50%. Additional training samples were generated using mirroring and random image augmentations: in-plane rotations between $\pm 20^{\circ}$, scale adjustments by $\pm 25\%$ and translations by $\pm 10\%$ of the bounding box size. Adam optimization [29] was used with an initial learning rate $\alpha = 0.001$ without decay and a batch size of 64 images. The first couple of epochs of the training process with the corresponding training and validation errors can be seen in Figure 4.3a.

The three subsequent stages were trained sequentially using the output of the previous stages as the initial positions. Local binary features and neural networks are trained as explained in section 4.2 in a two-stage process. The decision tree ensembles are trained for each landmark individually with the following parameters: tree depth $d = 5$, number of trees $n = 5$, the pool size to choose the best split parameters from is set to 512, the shrinkage factor for gradient boosting is set to $\nu = 0.5$, and the local region is set to the following percentages of the face scale for each stage: 16%, 10%, 10%. The neural networks are trained using SGD with a learning rate set to $\alpha = 1$ and momentum set to $\mu = 0.9$. The size of the bottleneck layer is gradually increased with each stage: $r = 40, 48, 56$. The reduction of the training and validation errors can be seen in Figure 4.3b as the optimization of the first local stage progresses.

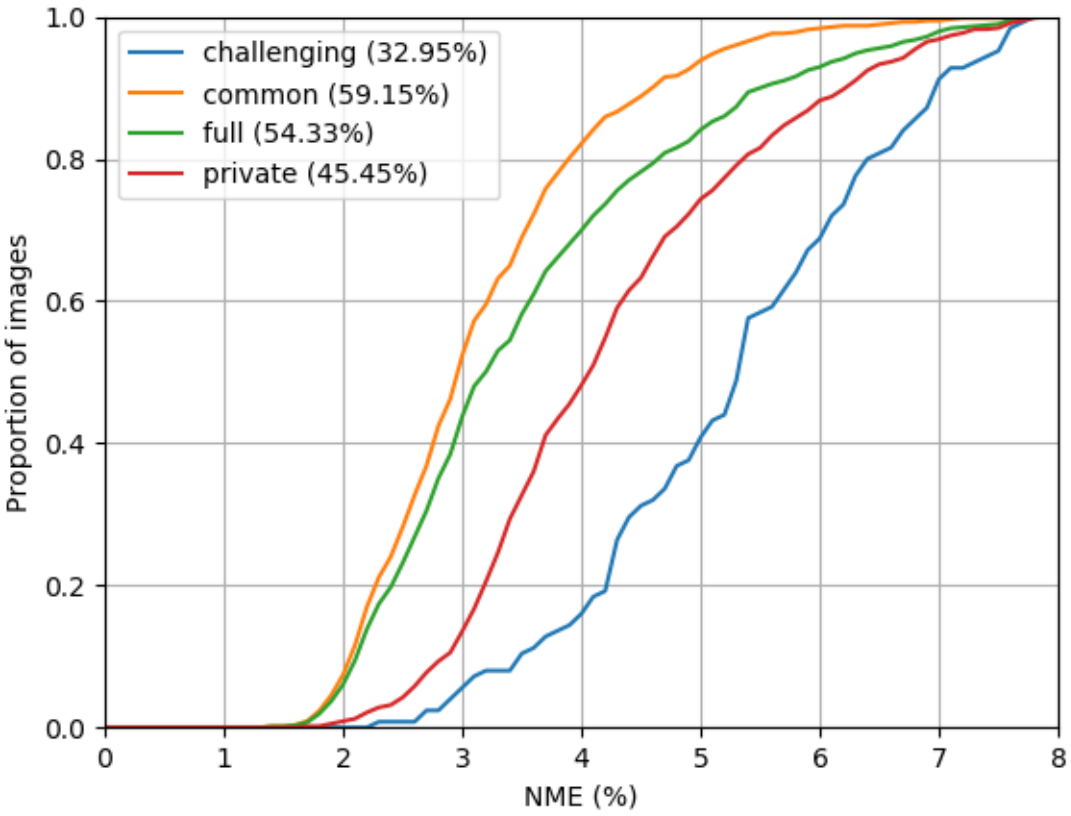


Figure 4.4: CED curves for each partition of the 300-W test set with corresponding $AUC_{0.08}$ measures using IOD normalization.

4.3.2 Results on 300-W

The CED curves and corresponding $AUC_{0.08}$ for each test partition using IOD normalization are shown in Figure 4.4. The worst performance is achieved on the challenging subset ($AUC_{0.08} = 32.95\%$), which is also the most difficult one, while the highest accuracy is achieved on the easiest, common subset ($AUC_{0.08} = 59.15\%$). The private test set proved to be more difficult than the full public test set with $AUC_{0.08} = 45.45\%$ and $AUC_{0.08} = 54.33\%$. Additionally, the failure rates for the full public test set $FR_{0.08} = 1.45\%$ and private test set $FR_{0.08} = 5.17\%$ also demonstrate the difference in difficulty between these two sets. On the other side, the achieved failure rates show a high degree of robustness of the proposed method having such a small proportion of samples considered failures. This is also visible from Figure 4.5 where nine samples with the worst errors are displayed, actually showing good robustness to difficult poses, illumination, and facial hair.

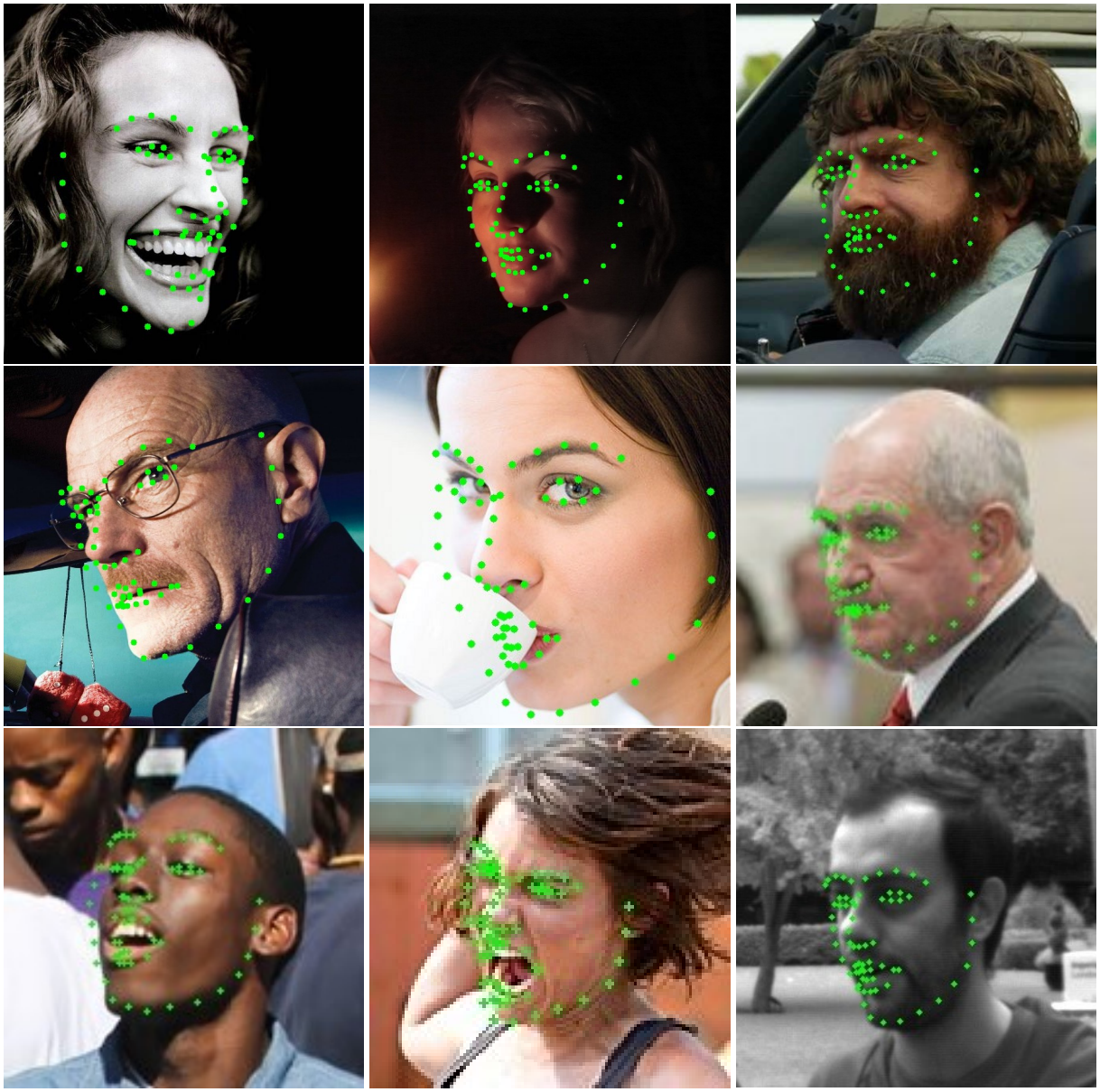


Figure 4.5: Samples from the 300-W challenging subset with the *highest* normalized mean error.

Comparison

Tables 4.2 and 4.3 show the achieved NME, AUC, and FR of the proposed method in comparison to previous work using IPD and IOD normalizations, respectively. The best results are achieved by heatmap regression methods using stacked hourglass architecture [93, 95, 96, 98] and methods utilizing additional training data [81, 89, 91, 93, 96]. The two top-performing algorithms from Yang et al. [93] and Liu et al. [96] used two face detection algorithms that also output a subset of landmarks to pre-align the faces for the stacked hourglass architecture training, making the alignment task less complex. The core contribution of the DVLN algorithm [81] is the use of additional training data (leveraging different mark-ups in data sets) from which every deep learning method should benefit. From the regression architecture stand-point, the stacked hourglass model demonstrates impressive results being featured in four out of five best-performing methods. However, it suffers from a high computational burden making real-time performance unfeasible even with a high-end GPU. Finally, there are two methods (Wing [89] and AWing [98]) among the top performers confirming the effectiveness of a customized loss compared to the standard L2 loss.

Competitive results on both challenging and common subsets without using external data are achieved by lightweight DCFE [63] algorithm, which utilizes global feature initialization through a CNN in combination with local cascaded regression (ERT [54]). The next three methods (FEC-CNC [72], DCR [66], and DAN [88]) use a combination of cascaded regression and deep learning to achieve good results but with a significant margin from the top performers. Methods DSRN [82] and TCDCN [124] both use a single deep learning model and external data to achieve results lagging from the leading methods, especially on the challenging subset. It seems that the coarse-to-fine approach from cascaded architectures is beneficial for deep learning models as well on such a complex task.

Traditional cascaded regression methods (LBF [15], SDM [55], and ESR [45]), although revolutionary at their time and extremely efficient, can not compete with the high capacity and flexibility of convolutional networks. The greatest problem for these methods and their derivatives is challenging images with ground truth far away from the initial shape. Additionally, 3D alignment methods [111, 115, 116] also struggle on 2D benchmarks. Although highly robust, as demonstrated on the challenging portion of the test set, disproportionately large errors are reported on the common partition. It is presumably due to the constraints imposed by the 3DMM as discussed in section 3.2.

The proposed method achieves comparable accuracy to other similar methods combining cascaded regression and deep learning [66, 72, 88], however, with a significantly lower computational complexity as explained in section 4.3.3. The failure rate is a bit more intuitive measure when evaluating face alignment accuracy. We can see that the two highly competitive methods DCFE [63] and DeCaFA [91] report $FR_{0.08} = 1.59\%$ and $FR_{0.1} = 0.15\%$, respectively. The

Table 4.2: Results reported on 300-W data set using IPD normalization. Methods marked with $*$ use external data for training.

Method	Common	Challenging	Full
RCPR [131]	6.18	17.26	8.35
ESR [45]	5.28	17.00	7.58
SDM [55]	5.57	15.40	7.50
3DDFA [111]	6.15	10.59	7.01
CFAN [87]	5.50	-	-
RPP [132]	5.50	11.57	6.69
ERT [54]	-	-	6.40
LBF [15]	4.95	11.98	6.32
iMORF [125]	-	-	6.31
Jourabloo et al. [115]	5.43	9.88	6.30
PRF [57]	4.90	11.96	6.28
DRDA* [134]	-	10.79	-
DeFA* [116]	5.37	9.38	6.10
cGPRT [59]	-	-	5.71
CFSS [79]	4.73	9.98	5.76
R-DSSD [61]	4.16	9.20	5.59
ERT-PIS [62]	4.42	10.32	5.58
KRFWS [64]	4.62	9.48	5.57
MCNet [84]	-	8.87	-
TCDCN* [124]	4.80	8.60	5.54
DSRN* [82]	4.12	9.68	5.21
DAN [88]	4.42	7.57	5.03
DCR [66]	4.19	8.42	5.02
FEC-CNC [72]	4.20	7.90	4.90
DCFE [63]	3.83	7.54	4.55
DVLN* [81]	3.79	7.15	4.45
AWing [98]	3.77	6.52	4.31
LAB [95]	3.42	6.98	4.12
Wing* [89]	**3.27**	7.18	4.04
Yang et al.* [93]	-	**7.0**	-
Liu et al.* [96]	3.45	6.38	**4.02**
Proposed	4.53	8.13	5.24

Table 4.3: Results reported on 300-W data set using IOD normalization. Methods marked with $*$ use external data for training.

Method	Common	Challenging	Full	AUC_{thr}	FR_{thr}
LBF-NN [58]	4.08	10.30	5.26	-	-
P-DSC-CR* [67]	3.83	6.93	4.38	-	-
MDM [60]	-	-	4.05	$52.12_{0.08}$	$4.21_{0.08}$
KRFWS [64]	3.34	6.56	3.97	-	-
DAN [88]	3.19	5.24	3.59	$55.33_{0.08}$	$\mathbf{1.16_{0.08}}$
LAB [95]	2.98	5.19	3.49	-	-
DeCaFA* [91]	2.93	5.26	3.39	$66.10_{0.1}$	$0.15_{0.1}$
HRNet [99]	2.87	5.15	3.32	-	-
DCFE [63]	2.76	5.22	3.24	$\mathbf{60.13_{0.08}}$	$1.59_{0.08}$
Yang et al.* [93]	-	4.9	-	-	-
AWing [98]	**2.72**	**4.52**	**3.07**	-	-
Proposed	3.27	5.63	3.73	$54.33_{0.08}$	$1.45_{0.08}$

proposed method achieves a similar failure rate $FR_{0.08} = 1.45$ confirming a competitive degree of robustness. It has been known for some time that face alignment has been a solved problem in controlled environments. However, these results on benchmarks in the wild suggest that it is close to being solved in general also.

4.3.3 Computational performance analysis

The initial global stage of the proposed method achieves an execution time of 2.32 ms using Intel's optimized OpenVINO* inference engine. With a slight accuracy drop, the performance can be further improved using quantization and INT8 computation. The three local refinement stages utilizing fast LBF and neural network execute in a total of 0.29 ms. This efficiency comes from the sparse nature of the feature vector, which reduces the multiplication of the first neural network layer to a series of memory look-ups and additions. The total execution time, including the spatial transformation of the image between the global and local stages, equals 3.05 ms measured on a single core i7-7500U operating at 2.7GHz.

Table 4.4 presents execution times in milliseconds on CPUs and GPUs for methods that report them in their publications. It is immediately visible that all fast methods with frame rates

*`https://github.com/openvinotoolkit/openvino`

Table 4.4: Reported execution times in milliseconds.

Method	Device	Exec. time
iMORF [125]	Core i7 @ 3.6 GHz	350
RCPR [131]	CPU @ 3.47 GHz	333.33
RPP [132]	CPU @ 3.3 GHz	250
Jourabloo et al. [115]	GTX Titan X	232.56
P-DSC-CR [67]	-	100
FEC-CNC [72]	-	100
3DDFA [111]	GTX Titan Black	75.72
LAB [95]	GTX Titan X	60
FAN [92]	GTX Titan X	34
PRF [57]	Core i7-2600	33.33
DeCaFA [91]	GTX 1060	31.25
DCFE [63]	Xeon E5-1650 @ 3.5 GHz	31.25
R-DSSD [61]	Core i5-6500 @ 3.2 GHz	25
DAN [88]	GTX 1070	22.22
TCDCN [124]	Core i5	18
MCNet [84]	Core i5-6200U	18
DVLN [81]	Core i5-4300u	15.15
SDM [55]	Core i7-2600	14.3
cGPRT [59]	Core i5-3570 @ 3.4 GHz	10.75
ESR [45]	Core i7-2600	8.34
Wing [89]	GTX Titan X	5.88
ERT-PIS [62]	Core i5-3470 @ 3.2 GHz	4.48
LBF [15]	Core i7-2600	3.12
DSRN [82]	GTX 1080Ti	2
LBF-NN [58]	Core i7-2600 @ 3.4 GHz	1.43
ERT [54]	-	1
Proposed	Core i7-7500U @ 2.7 GHz	3.05

above 60 FPS on a single CPU core utilize simple and fast features such as PDF or SIFT. The fastest method reporting 1 ms execution time is the ERT [54] method which uses PDF and fast decision trees. Other similar methods such as LBF [15] and LBF-NN [58] also report similar times with additional lighter and hyper-fast versions operating at above 3000 FPS. These methods, however, demonstrate significantly lower robustness and accuracy, unlike the proposed method, which is both fast and robust.

Any method utilizing large CNNs without optimizations can achieve real-time performance only by employing high-end GPU hardware. Notable examples are DAN [88] and FAN [92] algorithms with execution times of 22.22 ms and 34 ms, respectively. The recent heatmap regression methods are incapable of real-time performance even on a GPU and, thus, mostly do not report execution time except LAB [95] (60 ms). However, recent work in [82] (DSRN) demonstrates an impressive execution time of 2 ms on a GPU by utilizing more efficient convolutional layers [83]. Another example of a CNN architecture optimization, this time by reducing the complexity of the problem and thus required CNN complexity, is demonstrated in DVLN [81] with a reported execution time of 15.15 ms on a CPU. A method similar to the proposed with both competitive accuracy and real-time performance is DCFE [63] reporting an execution time of 31.25 ms on a CPU.

4.4 Discussion

Taking into account accuracies on the benchmark data set and computational complexity, the proposed method demonstrates both high robustness and efficiency. This is best illustrated in Figure 4.6 where both execution time and accuracy are graphically compared with relevant previous work. Other competitive methods on both account include DSRN [82], Wing [89], DVLN [81], and DCFE [63]. Most of these methods utilize additional training data to improve accuracy and high-end GPUs to achieve real-time performance. A notable exception is DCFE [63] which uses a similar cascaded regression architecture and global initialization but with an order of magnitude slower execution time.

As future work, there are multiple ways to improve the proposed method. One architectural improvement is to use part-based fine-tuning, the key technique that sets DCFE [63] apart from other cascaded regression methods. The last stages in the DCFE cascade do not regress a single monolithic face shape. It is broken up into semantic facial parts consisting of landmarks relevant for that region (e.g., eyes, mouth, nose, etc.). Even though early face alignment attempts used a similar approach [47], the important difference here is that it is used at the end of the cascade with landmarks already close to the ground truth positions. It enables the method to accurately align asymmetrical facial expressions not seen in the training set due to a large number of possible part combinations.

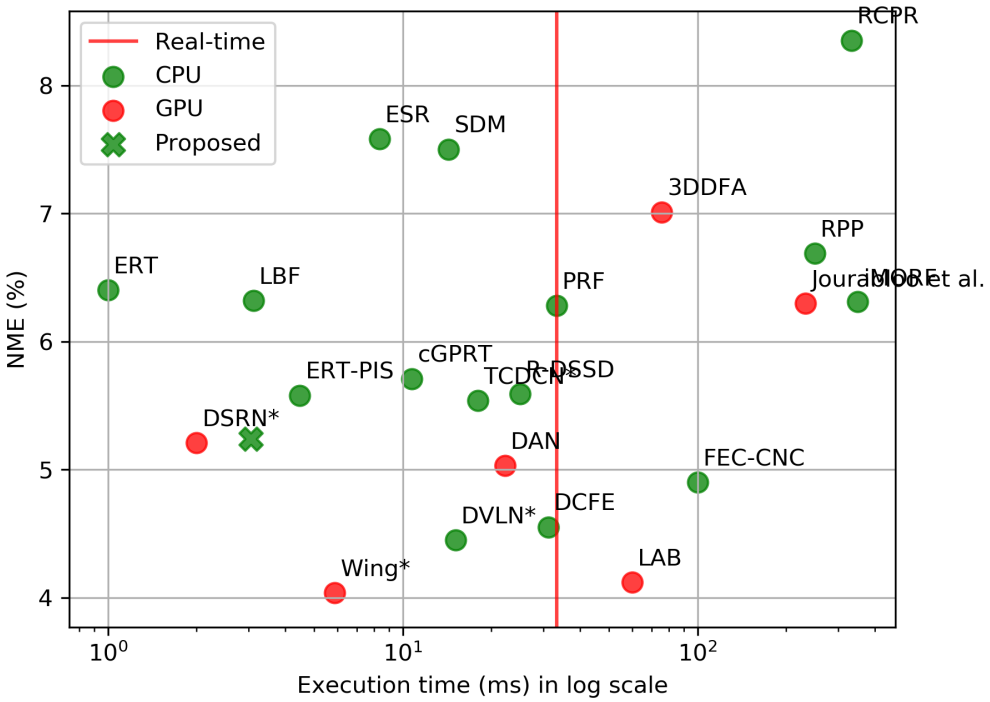


Figure 4.6: Comparison of both execution time and accuracy on 300-W full set using IPD normalization. Methods marked with $*$ use external data for training. The green cross marks the proposed method executing on a CPU. The green and red dots mark previous work executing on GPU and CPU, respectively.

In addition to the architectural improvements, it is evident from the comparisons that an increase in training data size improves accuracy, especially for methods utilizing deep learning [81, 93]. It is understandable since the size of the data sets is still quite small (300-W training set has 3148 images) due to the high complexity of the annotation process. Advanced data augmentation techniques are thus interesting to explore and use in combination with highly efficient algorithms, including synthetically generated images [194] and image warping to increase head pose variation [111]. Generative models have recently exploded in the research community with their ability to generate artificial photo-realistic images. One of the useful applications is the automatic creation of large-scale data sets, which would be especially beneficial for face alignment. Another promising use of generative models already being researched is for face alignment under heavy occlusions with the potential to "see" the parts of the face behind the obstacle.

Since face alignment is closely related to face detection, expressions, age, gender, and other face analysis tasks, it makes sense to unify the predictive models under a holistic approach. Past research confirms the merits of multi-task learning for related problems. The largest obstacle, however, is the unification of divergent data sets with different annotations. As already mentioned, generative models could be used to produce a well-balanced large-scale data set to train

a holistic face analysis model.

Nevertheless, the proposed method achieves a competitive accuracy with high computational efficiency, which is becoming increasingly important since both face tracking and facial expression recognition in videos are dynamic tasks with many applications requiring low latency. Having landmarks from salient facial regions efficiently localized, we can now proceed with facial expression recognition.

Chapter 5

Facial expression recognition using local binary features and shallow neural networks

The proposed method aims to identify six prototype facial expressions (anger, disgust, fear, happiness, sadness, and surprise) [1] from a single static 2D image. The method uses appearance-based features due to greater robustness to face shape variations when compared to geometric-based ones [7] with two key components as discussed in the following paragraphs.

Local feature learning As with many other computer vision problems, deep learning methods are taking precedence over traditional approaches using hand-crafted features for facial expression recognition (see section 3.5). The adoption, however, is slower due to low data availability. Our approach mitigates this problem using decision trees trained to extract relevant features efficiently using a low sample count. An additional advantage is the use of detected landmarks to extract local features from salient regions of the face [7, 8] further improving efficiency on small data sets.

Joint expression classification The advantage of joint classification lies in the fact that the expressions are not independent of each other. Some of them can be combined, also sharing facial muscle movements, while some are mutually exclusive, hinting at complex underlying relations. We propose a shallow neural network architecture including non-linearities to model these relations and improve classification accuracy.

As already mentioned, appearance features are extracted around facial landmarks (e.g., mouth and eyes); therefore, the first step is to detect the face and its landmarks. The face alignment method described in the previous chapter can be used, offering an excellent balance between accuracy and computational efficiency. The overview of the whole system is depicted

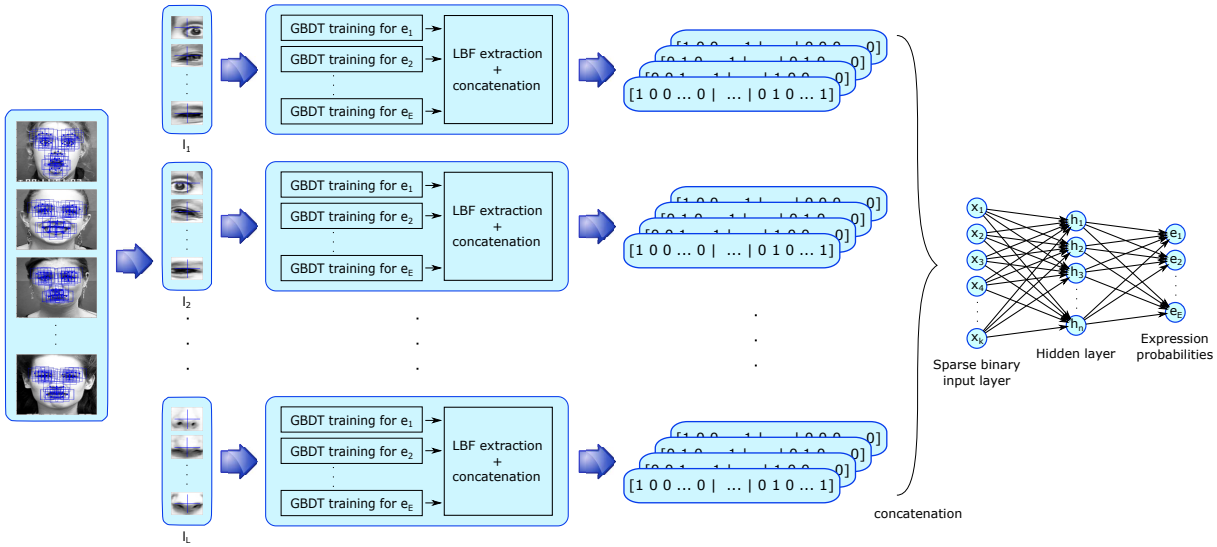


Figure 5.1: The proposed method takes an image of a face with detected landmark points. Local patches are used to train the gentle boosted decision trees for each expression in a one-vs-all manner. The tree ensembles are encoded into local binary features, which are concatenated into a single sparse binary feature vector. The sparse feature vector is used as an input into a simple 2-layer neural network that outputs the expression probabilities.

in Figure 5.1. The proposed implementation of the discussed key components will be described with more details in the following two sections 5.1 and 5.2. The method is evaluated on all relevant benchmarks in section 5.3 and results discussed in section 5.4.

5.1 Local feature learning

The key concept of this paper is the task-specific learning process for feature extraction, which encodes highly discriminative texture patterns for each facial expression around the detected facial landmarks (Figure 5.2). Ensembles of gentle boost decision trees [195] are trained with pixel difference features indexed to facial landmarks in order to maximize the one-vs-all posterior probability for each expression e around each landmark l. The number of trees within an ensemble and tree depth are specified in advance.

Let E and L denote the number of basic facial expressions and landmark points, respectively. For each facial expression $e \in \{e_1, \ldots, e_E\}$, we train an ensemble of gentle boost decision trees around each landmark point $l \in \{l_1, \ldots, l_L\}$ as can be seen on the left side of Figure 5.1. Let C represent the sample patches of an expression e and landmark l at the decision tree node n. Each candidate split $\theta = (p_1, p_2, t_n)$ from a random pool of generated parameters, divides the training samples in the following way:

$$C_{left}(\theta) = I(p_1) - I(p_2) \leq t_n \tag{5.1}$$

$$C_{right}(\theta) = C \setminus C_{left}(\theta) \tag{5.2}$$

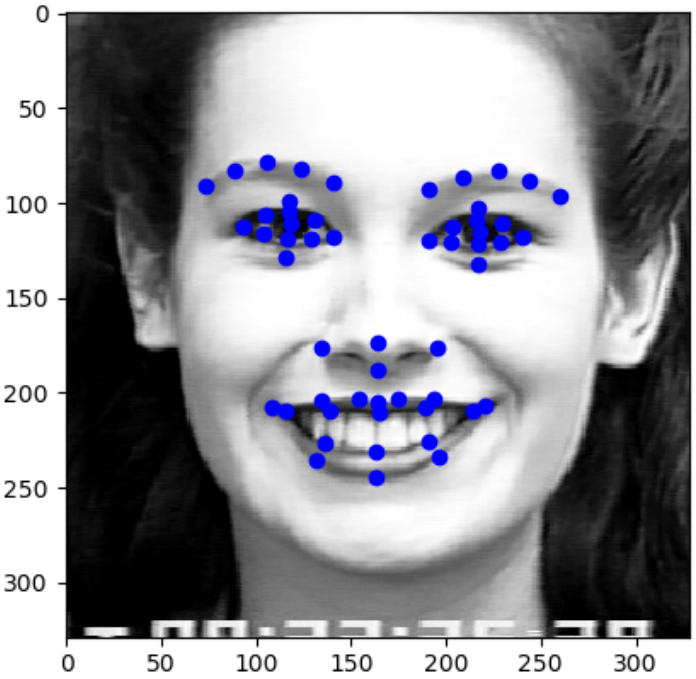


Figure 5.2: The detected landmark points used for LBF extraction regions.

where p_1 and p_2 represent the local patch positions, t_n represents the threshold and I represents the image intensities. The positions are placed relative to corresponding landmark location as depicted on the left part of Figure 5.3.

The cost function Q that is minimized consists of a Gini impurity measure:

$$G(X_n) = p_n(1 - p_n) \tag{5.3}$$

where p_n represents the proportion of expression e observations at node n:

$$p_n = \frac{1}{N_n} \sum_{x_i \in R_n} I(y_i = e) \tag{5.4}$$

R_n and N_n represent the sample space and number of samples at node n, respectively. y_i and x_i represent the current ground truth label (one-vs-all binary label) and sample patch, respectively. The full cost function is a weighted sum of impurity measures for both data partitions:

$$Q(C, \theta) = \frac{n_{left}}{N_n} G(C_{left}(\theta)) + \frac{n_{right}}{N_n} G(C_{right}(\theta)) \tag{5.5}$$

The described decision trees are organized into ensembles with the gentle boosting algorithm [195] in place. The algorithm ensures more emphasis is put on misclassified samples from the pre-

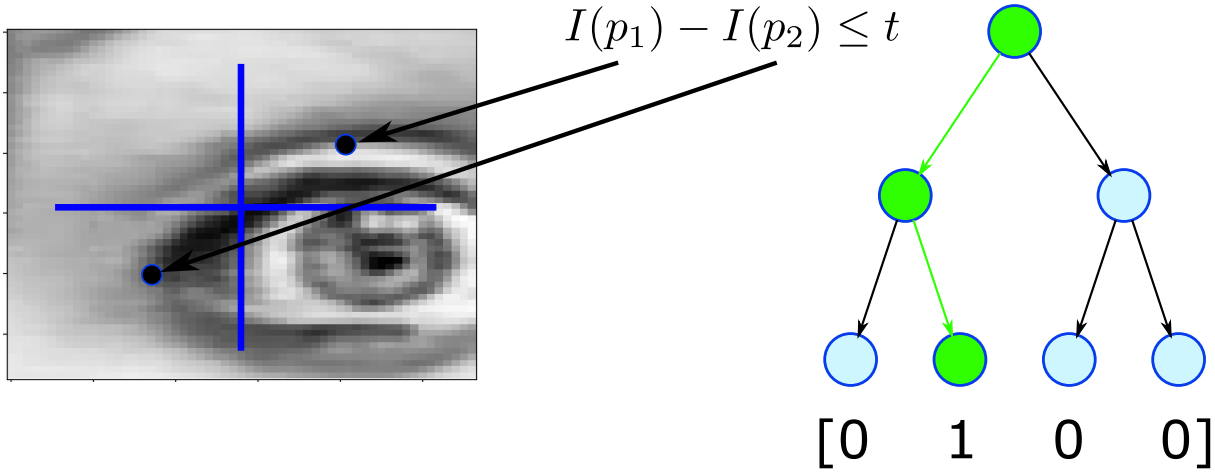


Figure 5.3: The decision trees use shape-indexed pixel difference features to split the training set. When encoding a sample into a local binary feature vector, a binary one is placed at the vector index corresponding to the leaf node where the sample ended up after traversing the tree.

vious tree in the ensemble. In practice, each sample i has a weight w_i assigned to it, which is increased or decreased depending on the output of the previous tree o_i:

$$w_i := w_i e^{-(y_i o_i)} \tag{5.6}$$

By doing this, each successive tree in the ensemble is forced to find even more discriminative features compared to the previous trees.

Once gentle boost ensembles for each facial expression and each landmark point are trained, local binary features are extracted as depicted in Figure 5.3. Each tree of an ensemble yields a tree vector of size equal to the number of leaves in that tree. All elements in that tree vector are equal to 0 except the one that corresponds to the leaf in which the given sample ended up while traversing that tree. This element is equal to 1. The tree vectors are concatenated into an ensemble vector with respect to the order of the trees. Each facial expression e gets ensemble-vectors $\phi_{\mathbf{e,l}}$ where $l \in \{l_1, \ldots, l_L\}$.

These ensemble vectors are concatenated to acquire a global binary feature vector Φ_e for each sample (Figure 5.1). It represents relevant pattern information for each expression:

$$\Phi_e = [\phi_{\mathbf{e,1}}, \ldots, \phi_{\mathbf{e,L}}]. \tag{5.7}$$

5.2 Expression classification

Feature vectors Φ_e for each expression e are concatenated into a single feature vector Φ which is used as an appearance-based representation of the face specifically tuned for expression dif-

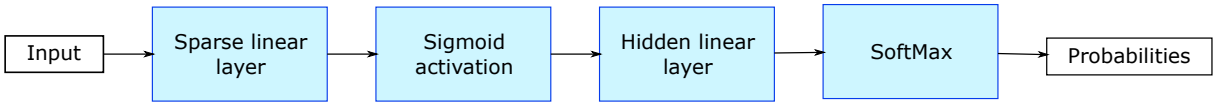


Figure 5.4: The diagram of the simple neural network architecture used to predict the expression probabilities.

ferentiation in a completely automatic supervised manner:

$$\Phi = [\Phi_1, \ldots, \Phi_E] \tag{5.8}$$

A shallow neural network with one hidden layer is used on the described sparse binary feature vector Φ. This simple network architecture (Figure 5.4) has demonstrated enough capacity to model the non-linear relationship between different expressions as shown in section 5.3.1. The network is trained using a cross-entropy criterion which is minimized over the data set:

$$\Theta^N = \arg\min_{\theta}(-\sum_{e=1}^{E} \log P(e)) \tag{5.9}$$

where $P(e)$ represents the probability of each expression e obtained by appending a soft-max layer at the end of the network:

$$P(e) = \frac{e^{x_e}}{\sum_{k=1}^{E} e^{x_k}} \tag{5.10}$$

The optimized network parameters Θ^N are obtained using a quasi-Newton method for optimization called Limited memory BFGS which approximates the Hessian matrix inverse when searching for the optimal descent direction [196]. Since all of the data sets are quite small, the whole training set is used in each iteration of the optimization. In order to improve the convergence speed, Wolfe conditions were used to modify the step length of the descent direction at each iteration [197].

5.3 Evaluation

We evaluated our system on the four most commonly used data sets for FER: CK+ [198], MMI [199], JAFFE [200], and SFEW 2.0 [201]. Due to the small size of the data sets, all of the experiments (except SFEW 2.0, which has a defined protocol) were conducted using a ten-fold cross-validation procedure which randomly divides the data sets into ten training and validation subsets. By doing this, every sample has both been in the training and validation set in one of the folds. The results were averaged across folds.

Furthermore, our experiments were strictly divided into person-independent (PI) and person-dependent (PD) scenarios. The PI scenario assures a strict subject division between the training

and validation sets, meaning the same person can not appear in both sets with different expressions. Naturally, the PI scenario is more complex; however, many researchers do not explicitly state their experimental procedure, making comparisons difficult. Both six and seven class results are reported since all of the data sets include a neutral expression also.

Face detection and alignment were first applied to all samples in the data sets. Since shape-indexed local features were used, no face registration and image transformations were needed as a preprocessing step. The only operation applied to the images was a conversion to gray-scale format since only pixel intensities are relevant and sampled by the decision trees.

5.3.1 Experiments on CK+

The Extended Cohn-Kanade (CK+) [198] data set is a widely recognized benchmark data set for FER. It contains 593 sequences from 123 subjects posing six prototypical expressions and contempt, additionally. All sequences start with a neutral expression and end with the peak of the requested expression. The peak frames are fully FACS annotated. Unlike other data sets, each expression label was verified using the FACS manual by certified FACS coders. Using the requested labels as the ground truth proved unreliable by the authors; thus, they added an additional validation step. After the validation, 327 of 593 sequences were determined to be of sufficient quality. Due to the comprehensiveness of the data set, we used it for the bulk of our experiments for parameter and architecture investigation.

According to the usual practice in static image FER, one neutral and three peak frames were used from each validated sequence. It amounts to the following number of samples per expression: 135 (An), 177 (Di), 75 (Fe), 207 (Ha), 84 (Sa), 249 (Su), 327 (Ne).

Decision tree parameters analysis

We explored the decision tree parameters (tree depth - TD and tree count in the ensembles - TC) using the PI scenario on the 7-class problem from the described CK+ data set. A simple logistic regression with a one-vs-all objective was used to train separate expression classifiers to set a baseline. Furthermore, the analysis using a simple logistic regression was suitable to narrow down the decision tree parameter space before analyzing the neural network architecture. The dimensionality of the final feature vector is calculated as follows:

$$D = 2^{TD} * TC * L * E \tag{5.11}$$

The tree parameters TC and TD directly affect the feature vector size, and since the dimensionality is quite high, regularization was needed to prevent over-fitting.

It is evident from Figure 5.5 that $TD = 2$ gives the overall best results regardless of the number of trees in the ensemble. Given the large dimensionality of the feature vector and the

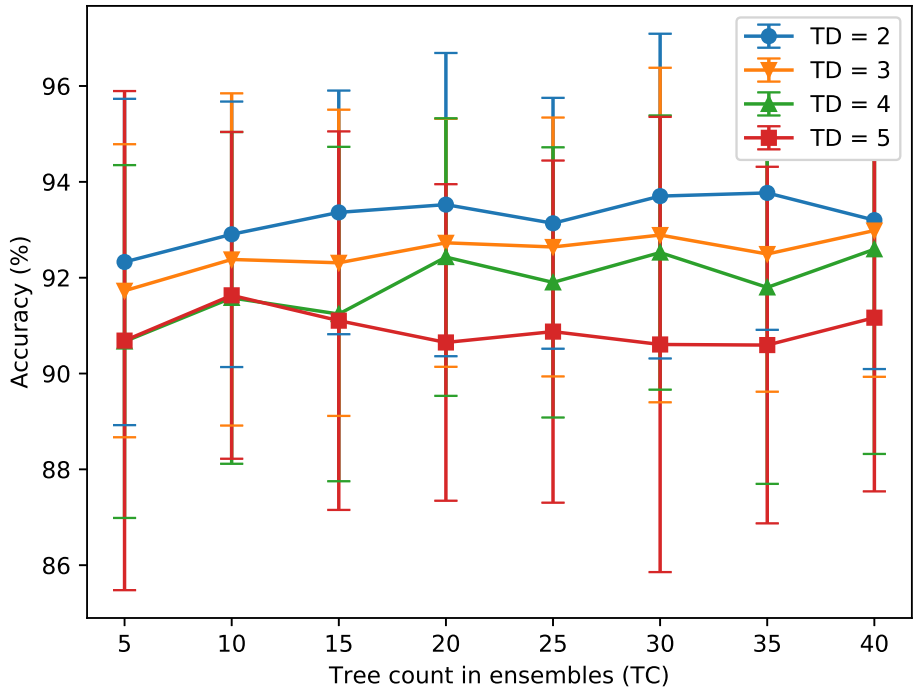


Figure 5.5: The accuracies and corresponding standard deviations plotted with error bars for different tree count TC and tree depth TD parameters trained with one-vs-all logistic regression on the PI scenario with seven classes from the CK+ data set.

relatively small size of the data set, it comes as no surprise that such simple trees are enough to capture relevant textural information. It is also clear from the graph that there is little or no added value in increasing the number of trees in the ensemble beyond 30. The best accuracy was achieved with $TD = 2$ and $TC = 35$, averaging 93.77%. We shall call this method LBF-LR.

Neural network parameters analysis

As already described, our neural network has one hidden layer whose size needed to be determined experimentally. We used the same scenario as in the previous section. We varied the size of the hidden layer HU while keeping decision tree parameters fixed to three configurations with the same three depth $TD = 2$: $TC = 20$, $TC = 25$, and $TC = 30$.

The results can be seen in Figure 5.6 where the optimal configuration is visible for parameters $TD = 2$, $TC = 25$ and $HU = 48$. When compared with the separate optimization using logistic regression from section 5.3.1, there is a boost in accuracy from 93.77% to 96.48%, which demonstrates the need for joint optimization to recognize facial expressions. We shall call this method LBF-NN.

Upon closer examination of the confusion matrices for both LBF-LR and LBF-NN showed in Figures 5.7a and 5.7b, we can see that the most important accuracy boosts are obvious for

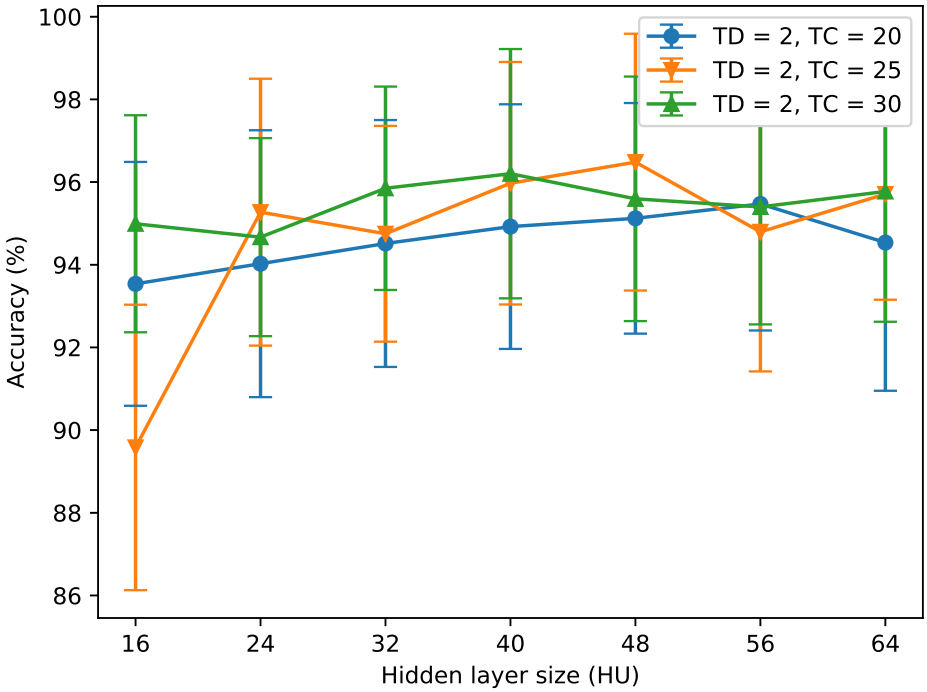
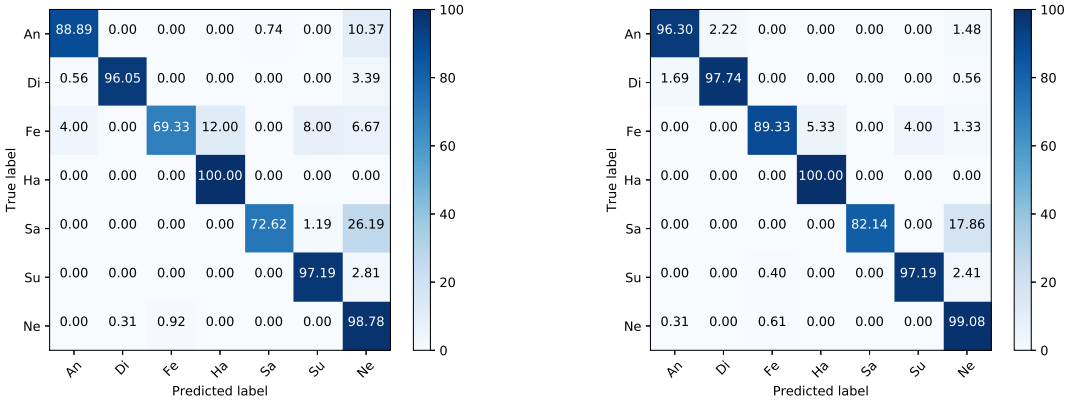


Figure 5.6: The accuracies and corresponding standard deviations plotted with error bars for different hidden layer sizes with selected decision tree configurations trained with the described neural network on the PI scenario with seven classes from the CK+ data set.

the most difficult expressions: fear and sadness. Incidentally, these two expressions have the least amount of samples in the data set due to the difficulty of truthfully portraying these emotions. Having a joint non-linear optimization process, features from other expressions can prove complementary and helpful to increase the recognition rate for these difficult expressions. The recognition rate increase for fear is 20%, while for sadness is 9.52%.

Comparison

It is quite difficult to compare our results to previous work since there is no official protocol described for the CK+ data set. We conducted experiments on both six- and seven-class (including neutral expression) problems with PD and PI scenarios using the best configuration described in section 5.3.1. The confusion matrices for the PI scenario are shown in Figures 5.8 and 5.7b. It is clear that the PD scenario is an easier task producing accuracies of 99.89% and 99.68% when compared to the PI scenario with accuracies of 98.08% and 96.48% for six- and seven-class problems, respectively. It is, therefore, very important to clearly and explicitly state the protocol of the experiments when comparing to other works. Upon closer inspection of the confusion matrices, we can see that by introducing the neutral expression, the overall recognition rate drops due to confusion between sadness and neutral expressions.



(a) LBF-LR (b) LBF-NN

Figure 5.7: The confusion matrix on the CK+ data set using seven classes and the PI scenario.

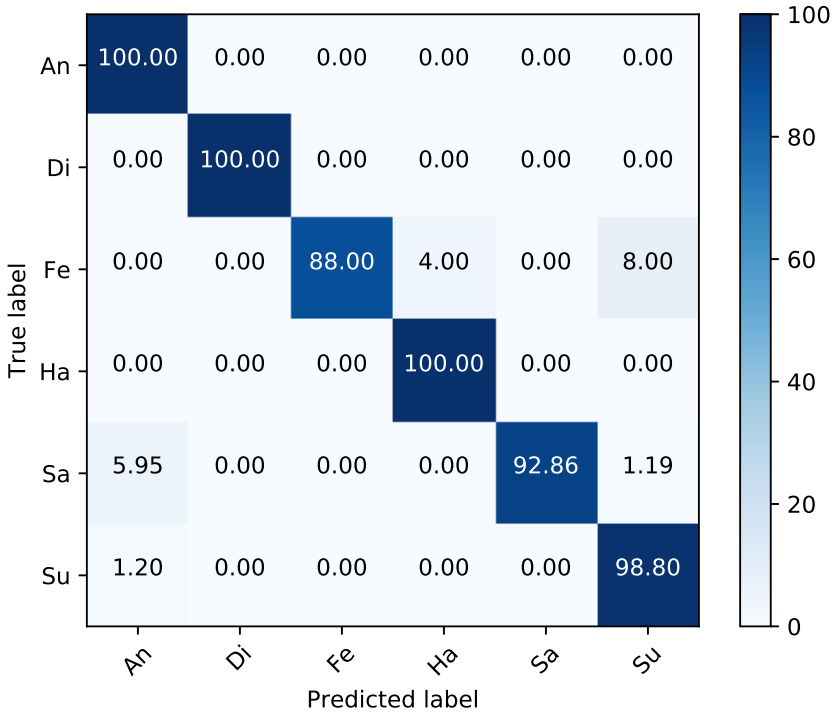


Figure 5.8: The confusion matrix for the proposed LBF-NN method on the CK+ data set using six classes with the PI scenario.

As we can see from Table 5.1, most of the previous methods differ in the number of classes, folds, and subjects used in the experiments. However, there is a positive trend of adopting the more difficult PI scenario. Our method is very competitive with other works for all experiment setups and sets a new state-of-the-art recognition rate for the CK+ data set with 96.48% for the seven-class problem. The previous best result was from Lopes et al. [163] where a CNN was used with various preprocessing methods to artificially increase the training set size and prevent over-fitting. The nature of our simpler LBF features makes it easier to train on smaller data sets and proves to be a viable alternative to heavy-weight convolutional features. Similarly, the current state-of-the-art method for the six-class problem uses a trained face recognition network to regularize and prevent DCNN expression over-fitting [168].

5.3.2 Results on MMI

MMI [199] data set contains more than 2900 videos and images of 75 subjects. It is an ongoing work to provide large volumes of data of facial expressions to the research community. Along with six basic emotions, it also contains single FACS Action Unit activation samples and naturalistic expressions. All of the videos include the starting neutral expression with the onset, apex, and offset phases. The major problem is that the apex frames are not indexed; therefore, it is hard to compare since researchers manually choose the frames to include in the training and validation sets.

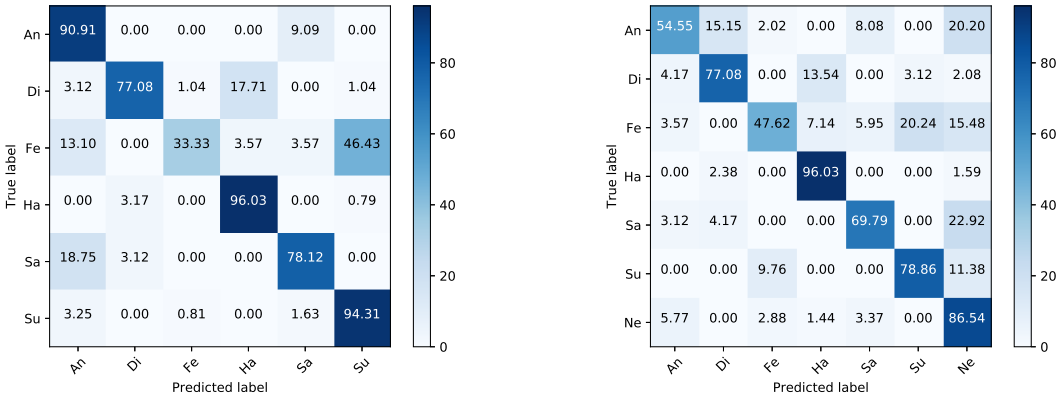
We filtered the data set to frontal view and seven basic expressions (including neutral), which resulted in 208 sequences (one sequence was corrupted) and 31 subjects. One neutral frame and three manually selected apex frames were used, totaling the following number of samples per expression: 99 (An), 96 (Di), 84 (Fe), 126 (Ha), 96 (Sa), 123 (Su), 208 (Ne). Again, no preprocessing was applied to the images except for the gray-scale conversion and the face detection/alignment to find the facial landmarks used in our method.

Four experiments were conducted similarly to the CK+ experiments, including six- and seven-class recognition in both PI and PD scenarios. The confusion matrices for the PI scenario are presented in Figures 5.9a and 5.9b. Once again, the PD scenario was easily solved with 99.84% and 99.88% recognition rates for six- and seven-class problems, respectively. However, the PI scenario proved to be much more difficult with recognition rates of 78.88% and 73.73% with optimal parameters presented in Table 5.2. A small L_2 regularization coefficient was used on the seven-class problem in the PI scenario that helped prevent over-fitting.

There are several reasons for these results. First of all, the MMI data set is much more challenging than the CK+ data set due to a large age span between subjects (19-62 years) and the fact that many subjects wore accessories like glasses and hats. Secondly, the sequences were not filtered by expert annotators; therefore, there is no guarantee that challenging expressions, such as fear and sadness, were acted out correctly and consistently across subjects. It is evident from

Table 5.1: Comparison with previous work on the CK+ data set.

Method	No. of folds	No. of subjects	Scenario	No. of classes	Recognition Rate (%)
Boughrara et al. [202]	10	97	PI	6	96.66
Gritti et al. [149]	10	95	not stated	7	92.90
Gu et al. [147]	10	94	PI	7	91.51
Happy and Routray [7]	10	118	not stated	6	94.09
Khan et al. [153]	10	not stated	PI	6	96.70
Lee et al. [203]	118	118	PI	7 (contempt)	90.47
Zhong et al. [8]	10	96	not stated	6	89.89
Littlewort et al. [156]	90	90	PI	7	93.30
Lopes et al. [163]	8	100	PI	6	96.76
			PI	7	95.75
Zhang et al. [148]	10	109	PI	6	95.50
			PI	7	93.60
Poursaberi et al. [161]	10	not stated	PI	6	86.10
			PD	6	90.37
Zhang and Tjondronegro [157]	10	92	PI	6	94.48
Liu et al. [162]	8	118	PI	6	96.70
Shan et al. [150]	10	96	PI	6	95.10
			PI	7	91.40
Mollahosseini et al. [164]	5	not stated	PI	6	93.20
Zavaschi et al. [151]	10	not stated	PI	7	88.90
			PD	7	99.40
Rivera et al. [159]	10	118	PI	7 (contempt)	89.30
Burkert et al. [171]	10	210	PD	7 (contempt)	99.60
Ding et al. [168]	10	not stated	PI	6	**98.60**
Proposed LBF-NN	10	118	PI	6	98.08
			PI	7	**96.48**
			PD	6	**99.89**
			PD	7	**99.68**



(a) Six classes

(b) Seven classes

Figure 5.9: The confusion matrix for the the proposed LBF-NN method on the MMI data set with the PI scenario.

Table 5.2: Optimal parameters for the PI scenario on the MMI, JAFFE, and SFEW 2.0 data sets.

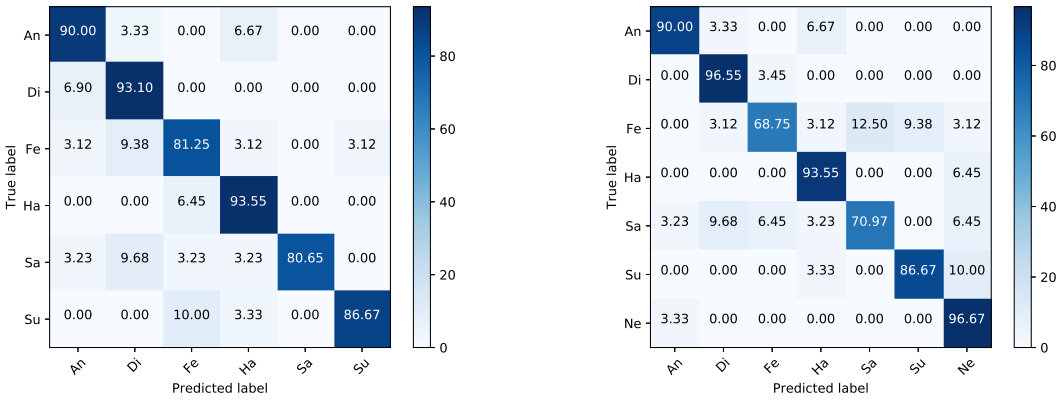
Data set	No. of classes	TD	TC	HU	L_2
MMI	6	2	20	16	0
MMI	7	2	25	24	0.0001
JAFFE	6	2	25	48	0
JAFFE	7	2	25	24	0
SFEW 2.0	7	2	30	24	0.0001

Table 5.3: Comparison with previous work on the MMI data set.

Method	No. of folds	No. of sequences/subjects	Scenario	No. of classes	Recognition rate (%)
Lee et al. [203]	20	150/21	PD	6	93.81
Zhong et al. [8]	10	205/not stated	not stated	6	77.39
Fang et al. [204]	10	203/not stated	not stated	6	75.96
Zhang et al. [148]	10	209/not stated	PI	6	93.60
			PI	7	**92.80**
Poursaberi et al. [161]	10	not stated	PI	6	86.10
			PD	6	90.37
Shan et al. [150]	10	96/20	PI	7	86.90
Mollahoseini et al. [164]	5	not stated/not stated	PI	6	77.60
Rivera et al. [159]	10	238/28	PI	6	**95.80**
Burkert et al. [171]	10	187/?	PD	6	98.63
Proposed LBF-NN	10	208/31	PI	6	78.88
			PI	7	73.73
			PD	6	**99.84**
			PD	7	**99.88**

the confusion matrices that it is very difficult to discern, e.g., fear from surprise and sadness from disgust. Thirdly, the results are very dependent on the peak frames used in the data set, which needed to be manually selected since the sequences are of varying length and different expression dynamics.

We compared ourselves with previous work in Table 5.3. Again, comparison on this data set is even harder since data acquisition is an ongoing process. Also, as can be seen from Table 5.3, there is a large variation in the number of subjects and sequences used for training and testing. Some of the authors manually discarded sequences with poorly acted expressions. The method from Zhang et al. [148] uses an almost identical set in their experiments and achieves a state-of-the-art recognition rate. However, they use hand-crafted features (fusion of LBPH and HOG) coupled with a multi-kernel SVM. Due to the hand-crafted features making their model less



(a) Six classes

(b) Seven classes

Figure 5.10: The confusion matrix for the proposed LBF-NN method on the JAFFE data set in the PI scenario.

complex, it is also less prone to over-fitting on small data sets. Another important point to note is that they fine-tuned their hyper-parameters on each fold in the cross-validation tests making the models highly specialized for combinations of specific fold training and test sets. Our tests were done with hyper-parameters optimized using the average accuracy across folds, not at the fold level. Furthermore, no cross-database experiments were conducted by the authors to test the generalization ability of their models. Another hand-crafted features method from Rivera et al. [159] achieves the state-of-the-art performance in the PI scenario with seven classes. The problem with comparing to this method is that only 168 sequences are available now from the 238 sessions they used.

5.3.3 Results on JAFFE

The Japanese Female Facial Expression database (JAFFE) [200] contains images of ten Japanese female models posing seven basic emotions. The total number of images is 213, making it the smallest data set we used for testing by far. An additional problem is that the data set obviously lacks diversity concerning gender, age, and race.

The same experiments were conducted as with the other two data sets and, similarly, the PD scenario recognition rates were extremely high above 98% for both six- and seven-class problems. However, as can be seen from the confusion matrices in Figures 5.10a and 5.10b, our method struggled again in the PI scenario to discern difficult and similar expressions such as fear and sadness. It can again be explained by the difficulty of sincerely portraying such emotions on demand. Nevertheless, in the easier six-class task, our method achieves recognition rates above 80% for each expression.

Table 5.4 compares our method to previous work on this data set. We achieve state-of-the-art results in the PD scenario due to the high flexibility of our method to adapt its feature extraction

Table 5.4: Comparison with previous work on the JAFFE data set.

Method	No. of folds	No. of images	Scenario	No. of classes	Recognition rate (%)
Gu et al. [147]	10	213	PI	7	89.67
Happy and Routray [7]	10	183	not stated	6	91.80
Lee et al. [204]	20	213	PD	6	94.70
Lopes et al. [163]	10	213	PI	6	53.44
			PI	7	53.57
Poursaberi et al. [161]	10	213	PI	7	91.12
			PD	7	95.04
Zhang and Tjondronegoro [157]	10	213	PI	6	92.93
Liu et al. [162]	10	213	PI	7	**91.80**
Shan et al. [150]	10	213	PI	7	81.00
Owusu et al. [158]	10	213	PD	6	96.83
Zavaschi et al. [151]	10	213	PI	7	70.00
			PD	7	96.20
Rivera et al. [159]	10	213	PI	6	**93.40**
			PI	7	90.60
Proposed LBF-NN	10	213	PI	6	87.22
			PI	7	85.88
			PD	6	**98.33**
			PD	7	**98.10**

process. In the PI scenario, we achieve competitive recognition rates of 87.22% and 83.56% for the six- and seven-class problems, respectively.

5.3.4 Results on SFEW 2.0

The Static Facial Expressions in the Wild (SFEW) [205] data set aims to benchmark the performance of FER methods in realistic conditions with unconstrained lighting, head poses, and occlusions. The second version of the data set we used in our experiments was released as part of the EmotiW 2015 challenge [201]. The images were extracted and annotated semi-automatically from movies and, even though the emotions are acted, the data set can be considered spontaneous since professional actors were involved.

The data set has a well-defined protocol with a strict division of training (958 images), validation (436 images), and test (372 images) sets. Since we could not obtain the labels for the test set, we report the results on the validation set only. The division of the data set is strictly person-independent. It contains seven basic expressions with the following number of samples (training and validation set combined): 255 (An), 89 (Di), 145 (Fe), 271 (Ha), 236 (Ne), 245 (Sa), 153 (Su).

Due to the unconstrained nature of the data set, we needed to modify the preprocessing pipeline to some extent. First, the face detector could not detect all of the faces, so we manually annotated eight images. Next, we used a more powerful face alignment method [92] that was trained on unconstrained head poses and can accurately align profile faces as well. Furthermore, we utilized the 2D landmark positions to remove the in-plane rotations of the faces, which reduced the variation of the relevant expression patterns around landmarks. Finally, we used horizontal mirroring to double the size of the training set. Even though this preprocessing step did not improve the results on other data sets, it proved beneficial here due to the asymmetry caused by large variations in head pose, illumination, and occlusions.

It is clear from the baseline results of the EmotiW 2015 challenge [201] (35.93% and 39.13% accuracy on validation and test sets, respectively) this is a very challenging benchmark. The optimal parameters for this data set are shown in Table 5.2 and the confusion matrix in Figure 5.11. It is evident once again that happiness is the easiest expression to recognize even in the unconstrained environment (80.82%); however, neutral and anger achieve respectable recognition rates as well (69.77% and 53.25%, respectively). Disgust and fear are traditionally very difficult to identify.

The proposed method achieves an average recognition rate of 49.31% without using any additional training data, which is the state-of-the-art result in such conditions. However, the best results are achieved by leveraging transfer learning with large related data sets (usually face recognition sets) and large ensembles of DCNNs [165, 167]. As seen from Table 5.5, all of the deep learning methods need auxiliary data sets and, even then, our method is very

Table 5.5: Comparison with previous work on the SFEW 2.0 data set.

Method	No. of images			Recognition Rate (%)		External data
	Train	Val	Test	Val	Test	
Zong et al. [160]	958	436	372	38.07	50.00	Yes
Mollahosseini et al. [164]	332	331	-	47.70	-	Yes
Ng et al. [166]	958	436	372	48.50	55.60	Yes
Zhai et al. [169]	958	436	-	48.51	-	Yes
Levi and Hassner [167]	891	431	372	51.75	54.56	Yes
Ding et al. [168]	891	431	-	55.15	-	Yes
Yu and Zhang [165]	958	436	371	**55.96**	**61.29**	Yes
Ding et al. [168]	891	431	-	48.19	-	No
Proposed LBF-NN	958	436	-	**49.31**	-	No

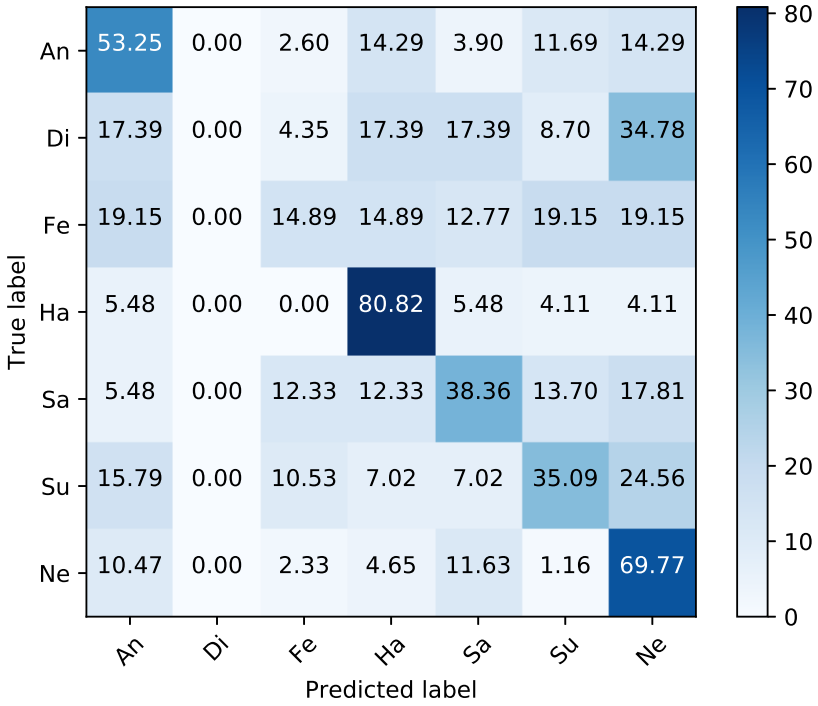


Figure 5.11: The confusion matrix for the proposed LBF-NN method on the SFEW 2.0 validation set.

Table 5.6: Comparison of cross-database recognition rates with seven classes.

Method	Train	Test	Recognition rate (%)
Zhang et al. [148]	CK+	MMI	**66.9**
	MMI	CK+	61.2
Shan et al. [150]	CK	MMI	51.1
Lee et al. [203]	MMI	CK+	64.57
Proposed LBF-NN	CK+	MMI	62.74
	MMI	CK+	**78.79**

competitive. The displayed results demonstrate the high robustness of the proposed method to unconstrained conditions. Furthermore, the method has once again shown an excellent ability to learn relevant information from a very limited amount of data.

5.3.5 Cross-database results

In order to test the generalization ability of our method, we conducted cross-database experiments with seven classes. We trained our method on CK+ and tested it on the MMI data set and vice versa. We chose these two databases because they have a similar number of samples and are at the opposite ends of the difficulty spectrum. The achieved results confirm these presumptions. When trained on the consistent and constrained CK+ data set and tested on the more challenging MMI set, we achieve the average recognition rate of 62.74%. When the situation is reversed, an impressive recognition rate of 78.79% is achieved. In fact, both results show a great generalization capacity of the proposed method since results in cross-database experiments are generally much worse than within database experiments.

It is interesting to observe here that the within database results for the MMI data set are *worse* (73.73%) than in the cross-database experiment with CK+ as the test set. This confirms the theory that the MMI data set is not consistently annotated and is quite difficult to train on. In Table 5.6 we compared our cross-database results with previous work, which provided similar experiments. Our method achieves the state-of-the-art result when generalizing from MMI to CK+ data set with an improvement of 14.22% from the previous best result.

Table 5.7: Comparison of computation time in milliseconds.

Method	CPU	Feature extraction	Classification	Total
Happy and Routray [7]	Intel i5 3.2 GHz	?	?	295.5
Khan et al. [153]	?	10	?	?
Lee et al. [203]	Pentium 3.50 GHz	110	40	150
Lopes et al. [163]	?	-	-	10
Zhang et al. [148]	Intel i5 2.66 GHz	?	30	?
Zhang and Tjondronegoro [157]	Core Duo 1.66 GHz	?	?	125.8
Liu et al. [162]	6-core 2.4 GHz	?	?	210
Shan et al. [150]	?	30	?	?
Owusu et al. [158]	?	?	?	14.5
Levi et al. [167]	Amazon GPU g2.8xlarge instance	?	?	500
Ding et al. [168]	Titan X GPU	?	?	3
Proposed LBF-NN	Intel i7-7500U 2.70 GHz	?	?	**1**

5.3.6 Computational performance analysis

We tested the recognition run-time of our method on a PC with an Intel Core i7-7500U CPU operating at 2.70 GHz frequency. The method is not parallelized and uses a single CPU core. The average computing time of our method on the JAFFE data set is approximately 1 ms which makes it ideal for mobile and embedded applications. Due to its simple pixel difference features coupled with shallow decision tree ensembles and a two-layer neural network, the online recognition phase is extremely efficient. The first neural network layer weight matrix is the largest one, and the multiplication with the large input feature vector would be the bottleneck of the system; however, due to the sparse binary nature of the feature vector, it can be computed with a simple series of memory lookups and additions. The run-time is written in C++, which contributes to fast execution.

We compared our method to previous work, which stated their execution time in Table 5.7. It is clear that our method achieves an order of magnitude improvement over all previous works. Ding et al. [168] achieve a real-time performance of 3 ms; however, they use a high-level GPU optimized code which is impractical for mobile and embedded systems.

5.4 Discussion

We presented a fast facial expression recognition method based on a trainable feature extraction process using ensembles of decision trees producing sparse binary feature vectors (LBF) and a shallow neural network. The two-layer neural network is capable of modeling the nonlinear relationship between expressions as demonstrated in section 5.3.1 which boosted the recognition rates of challenging expressions such as fear and sadness. The method uses static images and achieves state-of-the-art results on the most widely used CK+ database, demonstrates great generalization abilities in the cross-database experiments, and robustness on in-the-wild SFEW 2.0 data set. The high accuracy results are accompanied by an extremely fast computation time of 1 ms on a single CPU which is an order of magnitude improvement in speed compared to recent work. The accuracy and speed of the method make it ideal for FER in environments with limited resources such as embedded and mobile platforms. It is a viable alternative to end-to-end CNNs in scenarios with limited data sets and run-time resources.

Several factors contributed to the success of the proposed method. Unlike layers of trainable convolutional kernels used in deep learning methods, decision tree ensembles have demonstrated great generalization ability deduced from small data sets due to their simplistic nature. By limiting the possible feature space to local regions around prominent facial landmarks, their expressive power is further boosted, which resulted in highly discriminative and specialized features. Furthermore, joint classification with a shallow neural network exploited inter-class information, which contributed to the correct classification of ambiguous expressions.

As future work, the method could be extended to incorporate temporal information through the use of increasingly popular variants of Recurrent Neural Networks such as Long Short Term Memory (LSTM) networks. It would be natural since expressions are dynamic by nature, and their intensity changes over time. Another course of action would be to integrate occlusion and head pose information to make it more robust on in-the-wild images and videos.

Chapter 6

Conclusion

Efficient facial expression recognition is explored in this thesis using decision trees and neural networks. The first chapter introduces the problem of facial expression recognition along with its crucial sub-component facial landmark detection. The importance of both research fields is highlighted and the motivation for the thesis is established with an emphasis on computational efficiency. Since both problems are dynamic tasks, it is important to achieve real-time performance for low-latency applications, a natural fit for the combination of efficient decision trees and flexible neural networks. In the second chapter, decision trees are introduced as trainable feature extractors along with neural networks and deep learning, representing the building blocks for the novel algorithms presented in the fourth and fifth chapters. The third chapter presents an overview of related work for both problems, which serves as a reasoning background for the decisions made throughout the rest of the thesis. An efficient and robust face alignment method is introduced in the fourth chapter experimentally verified on the most widely used benchmark data set. A global CNN is used for robust initialization of the cascaded regression framework. The refinement stages use lightweight local binary features coupled with a bottleneck-based neural network architecture which improves both execution time and memory consumption. The detected landmark locations serve as a key input for the facial expression recognition method proposed in the fifth chapter. Expression-specific local binary features are extracted around each landmark for joint recognition using a shallow neural network architecture. The method achieves state-of-the-art accuracy on the CK+ data set with an order of magnitude margin in execution time compared to previous works.

The presented algorithms form a lightweight facial expression recognition system suitable for power-efficient devices with limited computational resources. Additionally, it is hard to collect large volumes of annotated data for some problems necessary for deep learning algorithms. The proposed combination of decision trees and neural networks demonstrates high reasoning capabilities on small-scale data sets and is a viable alternative to deep learning.

Bibliography

[1] Ekman, P., Friesen, W. V., "Constants across cultures in the face and emotion.", Journal of personality and social psychology, Vol. 17, No. 2, 1971, str. 124.

[2] Sandbach, G., Zafeiriou, S., Pantic, M., Yin, L., "Static and dynamic 3d facial expression recognition: A comprehensive survey", Image and Vision Computing, Vol. 30, No. 10, 2012, str. 683–697.

[3] Bettadapura, V., "Face expression recognition and analysis: the state of the art", arXiv preprint arXiv:1203.6722, 2012.

[4] Hjelmås, E., Low, B. K., "Face detection: A survey", Computer vision and image understanding, Vol. 83, No. 3, 2001, str. 236–274.

[5] Zeng, Z., Pantic, M., Roisman, G. I., Huang, T. S., "A survey of affect recognition methods: Audio, visual, and spontaneous expressions", IEEE transactions on pattern analysis and machine intelligence, Vol. 31, No. 1, 2009, str. 39–58.

[6] Mehrabian, A. *et al.*, Silent messages. Wadsworth Belmont, CA, 1971, Vol. 8.

[7] Happy, S., Routray, A., "Automatic facial expression recognition using features of salient facial patches", IEEE transactions on Affective Computing, Vol. 6, No. 1, 2015, str. 1–12.

[8] Zhong, L., Liu, Q., Yang, P., Huang, J., Metaxas, D. N., "Learning multiscale active facial patches for expression analysis", IEEE transactions on cybernetics, Vol. 45, No. 8, 2015, str. 1499–1510.

[9] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W., "Cosface: Large margin cosine loss for deep face recognition", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, str. 5265–5274.

[10] Jain, A. K., Li, S. Z., Handbook of face recognition. Springer, 2005, Vol. 1.

[11] Abate, A. F., Nappi, M., Riccio, D., Sabatino, G., "2d and 3d face recognition: A survey", Pattern Recognition Letters, Vol. 28, No. 14, 2007, str. 1885–1906.

[12] Booth, J., Roussos, A., Ponniah, A., Dunaway, D., Zafeiriou, S., "Large scale 3d morphable models", International Journal of Computer Vision, Vol. 126, No. 2-4, 2018, str. 233–254.

[13] Shen, J., Zafeiriou, S., Chrysos, G. G., Kossaifi, J., Tzimiropoulos, G., Pantic, M., "The first facial landmark tracking in-the-wild challenge: Benchmark and results", in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, str. 50–58.

[14] Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A., Classification and regression trees. CRC press, 1984.

[15] Ren, S., Cao, X., Wei, Y., Sun, J., "Face alignment at 3000 fps via regressing local binary features", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, str. 1685–1692.

[16] Breiman, L., "Random forests", Machine learning, Vol. 45, No. 1, 2001, str. 5–32.

[17] Friedman, J. H., "Greedy function approximation: a gradient boosting machine", Annals of statistics, 2001, str. 1189–1232.

[18] Ren, S., Cao, X., Wei, Y., Sun, J., "Global refinement of random forest", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, str. 723–730.

[19] McCulloch, W. S., Pitts, W., "A logical calculus of the ideas immanent in nervous activity", The bulletin of mathematical biophysics, Vol. 5, No. 4, 1943, str. 115–133.

[20] Werbos, P. J., "Applications of advances in nonlinear sensitivity analysis", in System modeling and optimization. Springer, 1982, str. 762–770.

[21] Glorot, X., Bengio, Y., "Understanding the difficulty of training deep feedforward neural networks", in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, str. 249–256.

[22] Fletcher, R., Practical methods of optimization. John Wiley & Sons, 2013.

[23] Liu, D. C., Nocedal, J., "On the limited memory bfgs method for large scale optimization", Mathematical programming, Vol. 45, No. 1-3, 1989, str. 503–528.

[24] Polyak, B. T., "Some methods of speeding up the convergence of iteration methods", USSR Computational Mathematics and Mathematical Physics, Vol. 4, No. 5, 1964, str. 1–17.

[25] Sutskever, I., Martens, J., Dahl, G., Hinton, G., “On the importance of initialization and momentum in deep learning”, in International conference on machine learning, 2013, str. 1139–1147.

[26] Duchi, J., Hazan, E., Singer, Y., “Adaptive subgradient methods for online learning and stochastic optimization.”, Journal of machine learning research, Vol. 12, No. 7, 2011.

[27] Tieleman, T., Hinton, G., “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”, COURSERA: Neural networks for machine learning, Vol. 4, No. 2, 2012, str. 26–31.

[28] Zeiler, M. D., “Adadelta: an adaptive learning rate method”, arXiv preprint arXiv:1212.5701, 2012.

[29] Kingma, D. P., Ba, J., “Adam: A method for stochastic optimization”, arXiv preprint arXiv:1412.6980, 2014.

[30] Hochreiter, S., Schmidhuber, J., “Long short-term memory”, Neural computation, Vol. 9, No. 8, 1997, str. 1735–1780.

[31] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., “Generative adversarial nets”, in Advances in neural information processing systems, 2014, str. 2672–2680.

[32] Elsken, T., Metzen, J. H., Hutter, F., “Neural architecture search: A survey”, Journal of Machine Learning Research, Vol. 20, No. 55, 2019, str. 1–21.

[33] Fukushima, K., Miyake, S., “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition”, in Competition and cooperation in neural nets. Springer, 1982, str. 267–285.

[34] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., “Backpropagation applied to handwritten zip code recognition”, Neural computation, Vol. 1, No. 4, 1989, str. 541–551.

[35] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., “Gradient-based learning applied to document recognition”, Proceedings of the IEEE, Vol. 86, No. 11, 1998, str. 2278–2324.

[36] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., “Imagenet: A large-scale hierarchical image database”, in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, str. 248–255.

[37] Krizhevsky, A., Sutskever, I., Hinton, G. E., "Imagenet classification with deep convolutional neural networks", in Advances in Neural Information Processing Systems, Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., (ur.), Vol. 25. Curran Associates, Inc., 2012, str. 1097–1105, dostupno na: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[38] Simonyan, K., Zisserman, A., "Very deep convolutional networks for large-scale image recognition", in International Conference on Learning Representations, 2015.

[39] He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, str. 770–778.

[40] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., "Going deeper with convolutions", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, str. 1–9.

[41] Gogić, I., Ahlberg, J., Pandžić, I. S., "Regression-based methods for face alignment: a survey", Signal Processing, 2020, str. 107755.

[42] Cootes, T. F., Edwards, G. J., Taylor, C. J., "Active Appearance Models", Proceedings of the European Conference on Computer Vision, Vol. 2, 1998, str. 484–498.

[43] Matthews, I., Baker, S., "Active Appearance Models Revisited", International Journal of Computer Vision, Vol. 60, No. 2, nov 2004, str. 135–164.

[44] Cootes, T., Taylor, C., Cooper, D., Graham, J., "Active Shape Models-Their Training and Application", Computer Vision and Image Understanding, Vol. 61, No. 1, jan 1995, str. 38–59.

[45] Cao, X., Wei, Y., Wen, F., Sun, J., "Face alignment by explicit shape regression", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012.

[46] Valstar, M., Martinez, B., Binefa, X., Pantic, M., "Facial point detection using boosted regression and graph models", Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010, str. 2729–2736.

[47] Kazemi, V., Sullivan, J., "Face Alignment with Part-Based Modeling", British Machine Vision Conference, 2011, str. 27.1–27.10.

[48] Bosch, A., Zisserman, A., Munoz, X., "Representing shape with a spatial pyramid kernel", in Proceedings of the 6th ACM international conference on Image and video retrieval. ACM, 2007, str. 401–408.

[49] Yang, H., Patras, I., "Face parts localization using structured-output regression forests.", in ACCV (2), 2012, str. 667–679.

[50] Yang, H., Patras, I., "Sieving regression forest votes for facial feature detection in the wild", in Proceedings of the IEEE International Conference on Computer Vision, 2013, str. 1936–1943.

[51] Martinez, B., Valstar, M. F., Binefa, X., Pantic, M., "Local evidence aggregation for regression-based facial point detection.", IEEE transactions on pattern analysis and machine intelligence, Vol. 35, No. 5, may 2013, str. 1149–63.

[52] Ojala, T., Pietikäinen, M., Mäenpää, T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 24, No. 7, 2002, str. 971–987.

[53] Dollár, P., Welinder, P., Perona, P., "Cascaded pose regression", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, str. 1078–1085.

[54] Kazemi, V., Sullivan, J., "One Millisecond Face Alignment with an Ensemble of Regression Trees", in 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, jun 2014, str. 1867–1874.

[55] Xiong, X., De La Torre, F., "Supervised descent method and its applications to face alignment", Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2013, str. 532–539.

[56] Lowe, D. G., "Distinctive image features from scale-invariant keypoints", International journal of computer vision, Vol. 60, No. 2, 2004, str. 91–110.

[57] Luo, C., Wang, Z., Wang, S., Zhang, J., Yu, J., "Locating facial landmarks using probabilistic random forest", Signal Processing Letters, IEEE, Vol. 22, No. 12, 2015, str. 2324–2328.

[58] Markuš, N., Gogić, I., Pandžić, I. S., Ahlberg, J., "Memory-efficient global refinement of decision-tree ensembles and its application to face alignment", in British Machine Vision Conference BMVC, 2018.

[59] Lee, D., Park, H., Yoo, C. D., "Face alignment using cascade Gaussian process regression trees", in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2015, str. 4204–4212.

[60] Trigeorgis, G., Snape, P., Nicolaou, M. A., Antonakos, E., Zafeiriou, S., "Mnemonic descent method: A recurrent process applied for end-to-end face alignment", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, str. 4177–4187.

[61] Liu, H., Lu, J., Feng, J., Zhou, J., "Learning deep sharable and structural detectors for face alignment", IEEE Transactions on Image Processing, Vol. 26, No. 4, 2017, str. 1666–1678.

[62] Zhu, H., Sheng, B., Shao, Z., Hao, Y., Hou, X., Ma, L., "Better initialization for regression-based face alignment", Computers & Graphics, Vol. 70, 2018, str. 261–269.

[63] Valle, R., Buenaposada, J. M., Valdes, A., Baumela, L., "A deeply-initialized coarse-to-fine ensemble of regression trees for face alignment", in Proceedings of the European Conference on Computer Vision (ECCV), 2018, str. 585–601.

[64] Kowalski, M., Naruniec, J., "Face alignment using k-cluster regression forests with weighted splitting", IEEE Signal Processing Letters, Vol. 23, No. 11, 2016, str. 1567–1571.

[65] Hara, K., Chellappa, R., "Growing regression forests by classification: Applications to object pose estimation", in European conference on computer vision. Springer, 2014, str. 552–567.

[66] Lai, H., Xiao, S., Cui, Z., Pan, Y., Xu, C., Yan, S., "Deep Cascaded Regression for Face Alignment", arXiv preprint arXiv:1510.09083, 2015.

[67] Liu, Q., Deng, J., Tao, D., "Dual Sparse Constrained Cascade Regression for Robust Face Alignment.", IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, Vol. 25, No. 2, feb 2016, str. 700–12.

[68] Aharon, M., Elad, M., Bruckstein, A., "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation", Signal Processing, IEEE Transactions on, Vol. 54, No. 11, 2006, str. 4311–4322.

[69] Yan, J., Lei, Z., Yi, D., Li, S., "Learn to combine multiple hypotheses for accurate face alignment", in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, str. 392–396.

[70] Feng, Z.-H., Huber, P., Kittler, J., Christmas, W., Wu, X.-J., "Random cascaded-regression copse for robust facial landmark detection", IEEE Signal Processing Letters, Vol. 1, No. 22, 2015, str. 76–80.

[71] He, Z., Zhang, J., Kan, M., Shan, S., Chen, X., "Robust fec-cnn: A high accuracy facial landmark detection system", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, str. 98–104.

[72] He, Z., Kan, M., Zhang, J., Chen, X., Shan, S., "A fully end-to-end cascaded cnn for facial landmark detection", in 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017). IEEE, 2017, str. 200–207.

[73] Zafeiriou, S., Trigeorgis, G., Chrysos, G., Deng, J., Shen, J., "The menpo facial landmark localisation challenge: A step towards the solution", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, str. 170–179.

[74] Xiong, X., De la Torre, F., "Global supervised descent method", in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2015, str. 2664–2673.

[75] Zhu, S., Li, C., Loy, C.-C., Tang, X., "Unconstrained face alignment via cascaded compositional learning", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, str. 3409–3417.

[76] Dong, Y., Wang, Y., Yue, J., Hu, Z., "Robust facial landmark localization using multi partial features", in 2015 IEEE International Conference on Image Processing (ICIP). IEEE, sep 2015, str. 98–102.

[77] Zhu, S., Li, C., Loy, C. C., Tang, X., "Towards arbitrary-view face alignment by recommendation trees", arXiv preprint arXiv:1511.06627, 2015.

[78] Rampal, K., Sakurai, K., Imaoka, H., "Occlusion handling in feature point tracking using ranked parts based models", in 2015 IEEE International Conference on Image Processing (ICIP). IEEE, sep 2015, str. 740–744.

[79] Shizhan Zhu, Cheng Li, Loy, C. C., Tang, X., "Face alignment by coarse-to-fine shape searching", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, str. 4998–5006.

[80] Calonder, M., Lepetit, V., Strecha, C., Fua, P., "Brief: Binary robust independent elementary features", in European conference on computer vision. Springer, 2010, str. 778–792.

[81] Wu, W., Yang, S., “Leveraging intra and inter-dataset variations for robust face alignment”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, str. 150–159.

[82] Miao, X., Zhen, X., Liu, X., Deng, C., Athitsos, V., Huang, H., “Direct shape regression networks for end-to-end face alignment”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, str. 5040–5049.

[83] Zhai, S., Cheng, Y., Zhang, Z. M., Lu, W., “Doubly convolutional neural networks”, in Advances in neural information processing systems, 2016, str. 1082–1090.

[84] Shao, Z., Zhu, H., Hao, Y., Wang, M., Ma, L., “Learning a multi-center convolutional network for unconstrained face alignment”, in 2017 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2017, str. 109–114.

[85] Sun, Y., Wang, X., Tang, X., “Deep Convolutional Network Cascade for Facial Point Detection”, in 2013 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, jun 2013, str. 3476–3483.

[86] “Extensive Facial Landmark Localization with Coarse-to-Fine Convolutional Network Cascade”, in 2013 IEEE International Conference on Computer Vision Workshops. IEEE, dec 2013, str. 386–391.

[87] Zhang, J., Shan, S., Kan, M., Chen, X., “Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment”, in Computer Vision–ECCV 2014. Springer, 2014, str. 1–16.

[88] Kowalski, M., Naruniec, J., Trzcinski, T., “Deep alignment network: A convolutional neural network for robust face alignment”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, str. 88–97.

[89] Feng, Z.-H., Kittler, J., Awais, M., Huber, P., Wu, X.-J., “Wing loss for robust facial landmark localisation with convolutional neural networks”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, str. 2235–2245.

[90] Jaderberg, M., Simonyan, K., Zisserman, A. *et al.*, “Spatial transformer networks”, in Advances in neural information processing systems, 2015, str. 2017–2025.

[91] Dapogny, A., Bailly, K., Cord, M., “Decafa: Deep convolutional cascade for face alignment in the wild”, in Proceedings of the IEEE International Conference on Computer Vision, 2019, str. 6893–6901.

[92] Bulat, A., Tzimiropoulos, G., "How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks)", in International Conference on Computer Vision, Vol. 1, No. 2, 2017, str. 4.

[93] Yang, J., Liu, Q., Zhang, K., "Stacked hourglass network for robust facial landmark localisation", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, str. 79–87.

[94] Chen, D., Hua, G., Wen, F., Sun, J., "Supervised transformer network for efficient face detection", in European Conference on Computer Vision. Springer, 2016, str. 122–138.

[95] Wu, W., Qian, C., Yang, S., Wang, Q., Cai, Y., Zhou, Q., "Look at boundary: A boundary-aware face alignment algorithm", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, str. 2129–2138.

[96] Liu, Z., Zhu, X., Hu, G., Guo, H., Tang, M., Lei, Z., Robertson, N. M., Wang, J., "Semantic alignment: Finding semantically consistent ground-truth for facial landmark detection", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, str. 3467–3476.

[97] Chen, L., Su, H., Ji, Q., "Face alignment with kernel density deep neural network", in Proceedings of the IEEE International Conference on Computer Vision, 2019, str. 6992–7002.

[98] Wang, X., Bo, L., Fuxin, L., "Adaptive wing loss for robust face alignment via heatmap regression", in Proceedings of the IEEE International Conference on Computer Vision, 2019, str. 6971–6981.

[99] Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X. *et al.*, "Deep high-resolution representation learning for visual recognition", IEEE transactions on pattern analysis and machine intelligence, 2020.

[100] Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K., "Facewarehouse: A 3d facial expression database for visual computing", IEEE Transactions on Visualization and Computer Graphics, Vol. 20, No. 3, 2013, str. 413–425.

[101] Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M. J., "A 3d facial expression database for facial behavior research", in 7th international conference on automatic face and gesture recognition (FGR06). IEEE, 2006, str. 211–216.

[102] Yin, L., Sun, X. C. Y., Worm, T., Reale, M., "A high-resolution 3d dynamic facial expression database, 2008", in IEEE International Conference on Automatic Face and Gesture Recognition, Amsterdam, The Netherlands, Vol. 126, str. 6.

[103] Besl, P. J., McKay, N. D., “Method for registration of 3-d shapes”, in Sensor fusion IV: control paradigms and data structures, Vol. 1611. International Society for Optics and Photonics, 1992, str. 586–606.

[104] Amberg, B., Romdhani, S., Vetter, T., “Optimal step nonrigid icp algorithms for surface registration”, in 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007, str. 1–8.

[105] Gerig, T., Morel-Forster, A., Blumer, C., Egger, B., Luthi, M., Schönborn, S., Vetter, T., “Morphable face models-an open framework”, in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, 2018, str. 75–82.

[106] Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y., Zafeiriou, S., “3d face morphable models" in-the-wild"”, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017, str. 5464–5473.

[107] Zhu, X., Lei, Z., Yan, J., Yi, D., Li, S. Z., “High-fidelity pose and expression normalization for face recognition in the wild”, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, str. 787–796.

[108] Tulyakov, S., Sebe, N., “Regressing a 3d face shape from a single image”, in Proceedings of the IEEE International Conference on Computer Vision, 2015, str. 3748–3755.

[109] Wu, Y., Ji, Q., “Shape augmented regression method for face alignment”, in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2015, str. 26–32.

[110] Zhao, R., Wang, Y., Benitez-Quiroz, C. F., Liu, Y., Martinez, A. M., “Fast and precise face alignment and 3d shape reconstruction from a single 2d image”, in European Conference on Computer Vision. Springer, 2016, str. 590–603.

[111] Zhu, X., Lei, Z., Liu, X., Shi, H., Li, S. Z., “Face alignment across large poses: A 3d solution”, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, str. 146–155.

[112] Deng, J., Zhou, Y., Cheng, S., Zaferiou, S., “Cascade multi-view hourglass model for robust 3d face alignment”, in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). IEEE, 2018, str. 399–403.

[113] Jourabloo, A., Liu, X., “Pose-invariant 3d face alignment”, in Proceedings of the IEEE International Conference on Computer Vision, 2015, str. 3694–3702.

[114] Jourabloo, A., Liu, X., “Pose-invariant face alignment via cnn-based dense 3d model fitting”, International Journal of Computer Vision, Vol. 124, No. 2, 2017, str. 187–203.

[115] Jourabloo, A., Ye, M., Liu, X., Ren, L., “Pose-invariant face alignment with a single cnn”, in Proceedings of the IEEE International Conference on computer vision, 2017, str. 3200–3209.

[116] Liu, Y., Jourabloo, A., Ren, W., Liu, X., “Dense face alignment”, in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, str. 1619–1628.

[117] Xie, S., Tu, Z., “Holistically-nested edge detection”, in Proceedings of the IEEE international conference on computer vision, 2015, str. 1395–1403.

[118] Bhagavatula, C., Zhu, C., Luu, K., Savvides, M., “Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses”, in Proceedings of the IEEE International Conference on Computer Vision, 2017, str. 3980–3989.

[119] Bookstein, F. L., “Principal warps: Thin-plate splines and the decomposition of deformations”, IEEE Transactions on pattern analysis and machine intelligence, Vol. 11, No. 6, 1989, str. 567–585.

[120] Jackson, A. S., Bulat, A., Argyriou, V., Tzimiropoulos, G., “Large pose 3d face reconstruction from a single image via direct volumetric cnn regression”, in Proceedings of the IEEE International Conference on Computer Vision, 2017, str. 1031–1039.

[121] Yu, R., Saito, S., Li, H., Ceylan, D., Li, H., “Learning dense facial correspondences in unconstrained images”, in Proceedings of the IEEE International Conference on Computer Vision, 2017, str. 4723–4732.

[122] Feng, Y., Wu, F., Shao, X., Wang, Y., Zhou, X., “Joint 3d face reconstruction and dense alignment with position map regression network”, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, str. 534–551.

[123] Evgeniou, T., Pontil, M., “Regularized multi–task learning”, in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004, str. 109–117.

[124] Zhang, Z., Luo, P., Loy, C. C., Tang, X., “Facial landmark detection by deep multi-task learning”, in Computer Vision–ECCV 2014. Springer, 2014, str. 94–108.

[125] Zhao, X., Kim, T.-K., Luo, W., “Unified Face Analysis by Iterative Multi-output Random Forests”, in 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, jun 2014, str. 1765–1772.

[126] Dantone, M., Gall, J., Fanelli, G., Van Gool, L., "Real-time facial feature detection using conditional regression forests", 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, str. 2578–2585.

[127] Chen, D., Ren, S., Wei, Y., Cao, X., Sun, J., "Joint cascade face detection and alignment", in European Conference on Computer Vision. Springer, 2014, str. 109–122.

[128] Zhang, K., Zhang, Z., Li, Z., Qiao, Y., "Joint face detection and alignment using multitask cascaded convolutional networks", IEEE Signal Processing Letters, Vol. 23, No. 10, 2016, str. 1499–1503.

[129] Deng, J., Trigeorgis, G., Zhou, Y., Zafeiriou, S., "Joint multi-view face alignment in the wild", IEEE Transactions on Image Processing, Vol. 28, No. 7, 2019, str. 3636–3648.

[130] Zhao, Y., Tang, F., Dong, W., Huang, F., Zhang, X., "Joint face alignment and segmentation via deep multi-task learning", Multimedia Tools and Applications, Vol. 78, No. 10, 2019, str. 13 131–13 148.

[131] Burgos-Artizzu, X. P., Perona, P., Dollar, P., "Robust face landmark estimation under occlusion", Proceedings of the IEEE International Conference on Computer Vision, 2013, str. 1513–1520.

[132] Heng Yang, Xuming He, Xuhui Jia, Patras, I., "Robust face alignment under occlusion via regional predictive power estimation.", IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, Vol. 24, No. 8, aug 2015, str. 2393–403.

[133] Wu, Y., Ji, Q., "Robust Facial Landmark Detection Under Significant Head Poses and Occlusion", in Proceedings of the IEEE International Conference on Computer Vision, 2015, str. 3658–3666.

[134] Zhang, J., Kan, M., Shan, S., Chen, X., "Occlusion-free face alignment: deep regression networks coupled with de-corrupt autoencoders", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, str. 3428–3437.

[135] Ekman, P., Friesen, W., "Facial action coding system: a technique for the measurement of facial movement. 1978", Consulting Psychologists, San Francisco.

[136] Jiang, B., Valstar, M., Martinez, B., Pantic, M., "A dynamic appearance descriptor approach to facial actions temporal modeling", IEEE transactions on cybernetics, Vol. 44, No. 2, 2014, str. 161–174.

[137] Jaiswal, S., Martinez, B., Valstar, M. F., "Learning to combine local models for facial action unit detection", in Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 6. IEEE, 2015, str. 1–6.

[138] Gudi, A., Tasli, H. E., den Uyl, T. M., Maroulis, A., "Deep learning based facs action unit occurrence and intensity estimation", in Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, Vol. 6. IEEE, 2015, str. 1–5.

[139] Sandbach, G., Zafeiriou, S., Pantic, M., "Markov random field structures for facial action unit intensity estimation", in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, str. 738–745.

[140] Jiang, B., Martinez, B., Valstar, M. F., Pantic, M., "Decision level fusion of domain specific regions for facial action recognition", in Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014, str. 1776–1781.

[141] Whitehill, J., Bartlett, M. S., Movellan, J. R., "Automatic facial expression recognition", Social Emotions in Nature and Artifact, Vol. 88, 2013.

[142] Huang, X., Zhao, G., Pietikäinen, M., Zheng, W., "Robust facial expression recognition using revised canonical correlation", in Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014, str. 1734–1739.

[143] Zhao, G., Pietikainen, M., "Dynamic texture recognition using local binary patterns with an application to facial expressions", IEEE transactions on pattern analysis and machine intelligence, Vol. 29, No. 6, 2007, str. 915–928.

[144] Guo, Y., Zhao, G., Pietikäinen, M., "Dynamic facial expression recognition with atlas construction and sparse representation", IEEE Transactions on Image Processing, Vol. 25, No. 5, 2016, str. 1977–1992.

[145] Rudovic, O., Pavlovic, V., Pantic, M., "Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation", in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, str. 2634–2641.

[146] Liu, M., Shan, S., Wang, R., Chen, X., "Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, str. 1749–1756.

[147] Gu, W., Xiang, C., Venkatesh, Y., Huang, D., Lin, H., "Facial expression recognition using radial encoding of local gabor features and classifier synthesis", Pattern Recognition, Vol. 45, No. 1, 2012, str. 80–91.

[148] Zhang, X., Mahoor, M. H., Mavadati, S. M., "Facial expression recognition using {l} _ {p}-norm mkl multiclass-svm", Machine Vision and Applications, Vol. 26, No. 4, 2015, str. 467–483.

[149] Gritti, T., Shan, C., Jeanne, V., Braspenning, R., "Local features based facial expression recognition with face registration errors", in Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on. IEEE, 2008, str. 1–8.

[150] Shan, C., Gong, S., McOwan, P. W., "Facial expression recognition based on local binary patterns: A comprehensive study", Image and Vision Computing, Vol. 27, No. 6, 2009, str. 803–816.

[151] Zavaschi, T. H., Britto, A. S., Oliveira, L. E., Koerich, A. L., "Fusion of feature sets and classifiers for facial expression recognition", Expert Systems with Applications, Vol. 40, No. 2, 2013, str. 646–655.

[152] Eleftheriadis, S., Rudovic, O., Pantic, M., "Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition", IEEE transactions on image processing, Vol. 24, No. 1, 2015, str. 189–204.

[153] Khan, R. A., Meyer, A., Konik, H., Bouakaz, S., "Framework for reliable, real-time facial expression recognition for low resolution images", Pattern Recognition Letters, Vol. 34, No. 10, 2013, str. 1159–1168.

[154] Dhall, A., Asthana, A., Goecke, R., Gedeon, T., "Emotion recognition using phog and lpq features", in Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on. IEEE, 2011, str. 878–883.

[155] Wan, S., Aggarwal, J., "Spontaneous facial expression recognition: A robust metric learning approach", Pattern Recognition, Vol. 47, No. 5, 2014, str. 1859–1868.

[156] Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J., Movellan, J., "Dynamics of facial expression extracted automatically from video", Image and Vision Computing, Vol. 24, No. 6, 2006, str. 615–625.

[157] Zhang, L., Tjondronegoro, D., "Facial expression recognition using facial movement features", IEEE Transactions on Affective Computing, Vol. 2, No. 4, 2011, str. 219–229.

[158] Owusu, E., Zhan, Y., Mao, Q. R., "A neural-adaboost based facial expression recognition system", Expert Systems with Applications, Vol. 41, No. 7, 2014, str. 3383–3390.

[159] Rivera, A. R., Castillo, J. R., Chae, O. O., "Local directional number pattern for face analysis: Face and expression recognition", IEEE transactions on image processing, Vol. 22, No. 5, 2013, str. 1740–1752.

[160] Zong, Y., Zheng, W., Huang, X., Yan, K., Yan, J., Zhang, T., "Emotion recognition in the wild via sparse transductive transfer linear discriminant analysis", Journal on Multimodal User Interfaces, Vol. 10, No. 2, 2016, str. 163–172.

[161] Poursaberi, A., Noubari, H. A., Gavrilova, M., Yanushkevich, S. N., "Gauss–laguerre wavelet textural feature fusion with geometrical information for facial expression identification", EURASIP Journal on Image and Video Processing, Vol. 2012, No. 1, 2012, str. 17.

[162] Liu, P., Han, S., Meng, Z., Tong, Y., "Facial expression recognition via a boosted deep belief network", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, str. 1805–1812.

[163] Lopes, A. T., de Aguiar, E., De Souza, A. F., Oliveira-Santos, T., "Facial expression recognition with convolutional neural networks: Coping with few data and the training sample order", Pattern Recognition, Vol. 61, 2017, str. 610–628.

[164] Mollahosseini, A., Chan, D., Mahoor, M. H., "Going deeper in facial expression recognition using deep neural networks", in Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on. IEEE, 2016, str. 1–10.

[165] Yu, Z., Zhang, C., "Image based static facial expression recognition with multiple deep network learning", in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015, str. 435–442.

[166] Ng, H.-W., Nguyen, V. D., Vonikakis, V., Winkler, S., "Deep learning for emotion recognition on small datasets using transfer learning", in Proceedings of the 2015 ACM on international conference on multimodal interaction. ACM, 2015, str. 443–449.

[167] Levi, G., Hassner, T., "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns", in Proceedings of the 2015 ACM on international conference on multimodal interaction. ACM, 2015, str. 503–510.

[168] Ding, H., Zhou, S. K., Chellappa, R., "Facenet2expnet: Regularizing a deep face recognition net for expression recognition", in Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on. IEEE, 2017, str. 118–126.

[169] Zhai, Y., Liu, J., Zeng, J., Piuri, V., Scotti, F., Ying, Z., Xu, Y., Gan, J., "Deep convolutional neural network for facial expression recognition", in International Conference on Image and Graphics. Springer, 2017, str. 211–223.

[170] Kim, B.-K., Dong, S.-Y., Roh, J., Kim, G., Lee, S.-Y., "Fusing aligned and non-aligned face information for automatic affect recognition in the wild: A deep learning approach", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, str. 48–57.

[171] Burkert, P., Trier, F., Afzal, M. Z., Dengel, A., Liwicki, M., "Dexpression: Deep convolutional neural network for expression recognition", arXiv preprint arXiv:1509.05371, 2015.

[172] Pramerdorfer, C., Kampel, M., "Facial expression recognition using convolutional neural networks: state of the art", arXiv preprint arXiv:1612.02903, 2016.

[173] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H. *et al.*, "Challenges in representation learning: A report on three machine learning contests", in International Conference on Neural Information Processing. Springer, 2013, str. 117–124.

[174] Long, M., Cao, Y., Wang, J., Jordan, M. I., "Learning transferable features with deep adaptation networks", arXiv preprint arXiv:1502.02791, 2015.

[175] Pan, S. J., Yang, Q. *et al.*, "A survey on transfer learning", IEEE Transactions on knowledge and data engineering, Vol. 22, No. 10, 2010, str. 1345–1359.

[176] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv preprint arXiv:1704.04861, 2017.

[177] Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H., "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation", in Proceedings of the european conference on computer vision (ECCV), 2018, str. 552–568.

[178] Tan, M., Le, Q. V., "Mixconv: Mixed depthwise convolutional kernels", arXiv preprint arXiv:1907.09595, 2019.

[179] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., "Mobilenetv2: Inverted residuals and linear bottlenecks", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, str. 4510–4520.

[180] Hu, J., Shen, L., Sun, G., "Squeeze-and-excitation networks", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, str. 7132–7141.

[181] Zhang, X., Zhou, X., Lin, M., Sun, J., "Shufflenet: An extremely efficient convolutional neural network for mobile devices", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, str. 6848–6856.

[182] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C., "Ghostnet: More features from cheap operations", in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, str. 1580–1589.

[183] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., Keutzer, K., "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size", arXiv preprint arXiv:1602.07360, 2016.

[184] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q. V., "Mnasnet: Platform-aware neural architecture search for mobile", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, str. 2820–2828.

[185] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. *et al.*, "Searching for mobilenetv3", in Proceedings of the IEEE International Conference on Computer Vision, 2019, str. 1314–1324.

[186] Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., Keutzer, K., "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, str. 10 734–10 742.

[187] Dai, X., Zhang, P., Wu, B., Yin, H., Sun, F., Wang, Y., Dukhan, M., Hu, Y., Wu, Y., Jia, Y. *et al.*, "Chamnet: Towards efficient network design through platform-aware model adaptation", in Proceedings of the IEEE Conference on computer vision and pattern recognition, 2019, str. 11 398–11 407.

[188] Tan, M., Le, Q., "Efficientnet: Rethinking model scaling for convolutional neural networks", in International Conference on Machine Learning, 2019, str. 6105–6114.

[189] Redmon, J., Farhadi, A., "Yolo9000: better, faster, stronger", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, str. 7263–7271.

[190] Ren, S., Cao, X., Wei, Y., Sun, J., "Face alignment via regressing local binary features", IEEE Transactions on Image Processing, Vol. 25, No. 3, 2016, str. 1233–1245.

[191] Reinsel, G. C., Velu, R. P., Multivariate Reduced-Rank Regression: Theory and Applications. Springer, 1998.

[192] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., Pantic, M., "300 faces in-the-wild challenge: The first facial landmark localization challenge", in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, str. 397–403.

[193] Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., Pantic, M., "300 faces in-the-wild challenge: Database and results", Image and vision computing, Vol. 47, 2016, str. 3–18.

[194] Feng, Z.-H., Hu, G., Kittler, J., Christmas, W., Wu, X.-J., "Cascaded Collaborative Regression for Robust Facial Landmark Detection Trained Using a Mixture of Synthetic and Real Images With Dynamic Weighting.", IEEE transactions on image processing : a publication of the IEEE Signal Processing Society, Vol. 24, No. 11, nov 2015, str. 3425–40.

[195] Friedman, J., Hastie, T., Tibshirani, R. *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)", The annals of statistics, Vol. 28, No. 2, 2000, str. 337–407.

[196] Byrd, R. H., Lu, P., Nocedal, J., Zhu, C., "A limited memory algorithm for bound constrained optimization", SIAM Journal on Scientific Computing, Vol. 16, No. 5, 1995, str. 1190–1208.

[197] Wolfe, P., "Convergence conditions for ascent methods", SIAM review, Vol. 11, No. 2, 1969, str. 226–235.

[198] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I., "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression", in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on. IEEE, 2010, str. 94–101.

[199] Pantic, M., Valstar, M., Rademaker, R., Maat, L., "Web-based database for facial expression analysis", in Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on. IEEE, 2005, str. 5–pp.

[200] Lyons, M., Akamatsu, S., Kamachi, M., Gyoba, J., "Coding facial expressions with gabor wavelets", in Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on. IEEE, 1998, str. 200–205.

[201] Dhall, A., Ramana Murthy, O., Goecke, R., Joshi, J., Gedeon, T., "Video and image based emotion recognition challenges in the wild: Emotiw 2015", in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. ACM, 2015, str. 423–426.

[202] Boughrara, H., Chtourou, M., Amar, C. B., Chen, L., "Facial expression recognition based on a mlp neural network using constructive training algorithm", Multimedia Tools and Applications, Vol. 75, No. 2, 2016, str. 709–731.

[203] Lee, S. H., Plataniotis, K. N. K., Ro, Y. M., "Intra-class variation reduction using training expression images for sparse representation based facial expression recognition", IEEE Transactions on Affective Computing, Vol. 5, No. 3, 2014, str. 340–351.

[204] Fang, H., Mac Parthaláin, N., Aubrey, A. J., Tam, G. K., Borgo, R., Rosin, P. L., Grant, P. W., Marshall, D., Chen, M., "Facial expression recognition in dynamic sequences: An integrated approach", Pattern Recognition, Vol. 47, No. 3, 2014, str. 1271–1281.

[205] Dhall, A., Goecke, R., Lucey, S., Gedeon, T., "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark", in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011, str. 2106–2112.

Biography

Ivan Gogić was born in Zagreb in 1985. He graduated from the Faculty of Electrical Engineering and Computing, the University of Zagreb, in 2009 with a concentration in control engineering and automation. In 2015, he started working as a research associate at the same institution on a research project funded by Visage Technologies. Currently, he works as a director of research and development in the Face Technology Division at Visage Technologies. He leads a team developing algorithms for face tracking and analysis, which are also his main scientific interests. He has published peer-reviewed papers in relevant scientific journals and in the proceedings of international conferences.

List of publications

Journal papers

1. **Gogić, I.**, Ahlberg, J., Pandžić, I.S., "Regression-based methods for face alignment: A survey", Signal Processing, Vol. 178, January 2021, doi:10.1016/j.sigpro.2020.107755.
2. **Gogić, I.**, Manhart, M., Pandžić, I.S., Ahlberg, J., "Fast facial expression recognition using local binary features and shallow neural networks", Visual Computer, Vol. 36, Iss. 1, August 2018, pp. 97-112, doi:10.1007/s00371-018-1585-8.
3. Bešenić K., **Gogić, I.**, Pandžić, I.S., Matković, K., "Automatic Image-based Face Analysis Systems Overview", Engineering Power : Bulletin of the Croatian Academy of Engineering, Vol. 13, Iss. 2, 2018, pp. 2-7, uri:https://hrcak.srce.hr/215882.

Conference publications

1. Markuš, N., **Gogić, I.**, Pandžić, I.S., Ahlberg, J., "Memory-efficient Global Refinement of Decision-Tree Ensembles and its Application to Face Alignment", Proceedings of the 29th British Machine Vision Conference, September 2018.

Životopis

Ivan Gogić rođen je u Zagrebu 1985. godine. Diplomirao je automatiku na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu 2009. godine. Zaposlio se kao znanstveni suradnik u istoj ustanovi 2015. godine na istraživačkom projektu u suradnji sa tvrtkom Visage Technologies. Trenutno je direktor razvoja i istraživanja u diviziji tehnologije lica tvrtke Visage Technologies. Vodi tim za razvoj algoritama praćenja i analize lica te su mu to ujedno i glavni znanstveni interesi. U tim područjima objavljuje radove u priznatim međunarodnim konferencijama i časopisima.