

UNIVERZITET U BEOGRADU

MATEMATIČKI FAKULTET

MASTER RAD

**Automatizacija pronalazjenja
mikroekspresija u video zapisima**

Autor:
Daniel ŠARANović

Mentor:
Dr. Mladen NIKOLIĆ

Katedra za računarstvo i informatiku

septembar, 2018.

Mentor: *Dr Mladen Nikolić, docent*
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

*Dr Filip Marić, vanredni
profesor*
Univerzitet u Beogradu,
Matematički fakultet

Dr Aleksandar Kartelj, docent
Univerzitet u Beogradu,
Matematički fakultet

Datum odbrane:

Potpis:

Sadržaj

1	Uvod	1
1.1	Mikroekspresije i njihove primene	1
1.2	Koraci implementacije alata za detekciju mikroekspresija	2
2	Implementacione tehnike	5
2.1	LBP Atributi	5
2.2	Detekcija i praćenje ključnih tačaka	8
2.2.1	Poravnati PDM	8
2.2.2	Smanjivanje dimenzionalnosti	9
2.2.3	Pronalaženje ključnih tačaka regularizovanim prosečnim pomerajima	10
2.2.3.1	Aproksimacija gausovim raspodelama	12
2.2.3.2	Regularizovani prosečni pomeraj	14
2.2.4	Eksperti polja	18
2.2.4.1	Učenje eksperta polja	19
2.3	Pronalaženje ekstremuma signala	23
2.4	Transformacija i izdvajanje lica	26
2.5	Lociranje mikroekspresija	27
3	Program za detekciju mikroekspresija	31
3.1	Nacrt programa	31
3.1.1	OpenFace biblioteka	32
3.2	Inicijalizacija parametara	33
3.3	Pretprocesiranje	35
3.4	Opis klasa za lociranje mikroekspresija	37
4	Rezultati programa	39
4.1	Opis baza podataka	39
4.1.1	CASME	40
4.1.2	CASME II	40
4.2	Eksperimentalni rezultati	41
4.2.1	Način procene tačnosti detektora	41
4.2.2	Rezultati za CASME II	41
4.2.3	Rezultati za CASME	42
4.3	Poređenje rezultata	43
4.3.1	CASME II	43
4.3.2	CASME	45
4.4	Analiza rezultata	47
4.4.1	CASME II	47
4.4.2	CASME	49
4.5	Performanse detektora	49

5	Pravci mogućih unapređenja	51
5.1	Metoda mašinskog učenja za proces detekcije mikroekspresija	51
5.2	Posebna analiza za treptaje oka	52
5.3	Računanje rastojanja u svrhu detekcije mikroekspresija	53
6	Zaključak	55
	Reference	57

Glava 1

Uvod

1.1 Mikroekspresije i njihove primene

Mikroekspresija je kratak, nekontrolisan izraz lica koji se javlja kod ljudi i koji sa sobom nosi informaciju o određenoj emociji. Mikroekspresije se javljaju kada osoba želi svesno da sakrije osećanja ili kada osoba nije svesna svojih osećanja. U najvećem broju slučajeva javljaju se kada osoba može nešto da dobije ili izgubi, a što je njoj od značaja.

Mikroekspresije su prvi put uočene od strane E. Hagarda i K. Isaaka 1966. godine [15] pri analizi snimaka psihoterapija pri traženju indikatora neverbalne komunikacije između pacijenta i terapeuta.

Paul Ekman je 1969. eksperimentalno pokazao da ne postoji nijedan pokret ruku ili nogu koji sam za sebe govori da li osoba laže [12]. Ipak postoje indikatori emocija (u vidu pokreta) koji ne moraju biti u skladu sa onim što osoba govori.

Mikroekspresije su jedan vid takvih indikatora. Veoma su kratke u trajanju (0.04s-0.5s) i slabe u intenzitetu. Veoma teško se voljno kontrolišu.¹ U početku, mikroekspresije su korišćene da bi se uočila obmana od strane pacijenta u raznim psihoterapeutskim razgovorima.

Kako je u velikom broju situacija veoma bitno znati da li su informacije tačne - razne agencije (CIA, FBI, ATF) kao i određene sudije, policijski službenici i advokati su podlegali obukama čiji je cilj bio tačnije prepoznavanje da li neko govori istinu ili ne [11]. Proces obuke sastojao se od toga da učesnici nauče da prepoznaju razne indikatore emocija, uključujući i mikroekspresije.

Određene osobe neće prikazati puno indikatora: glumci, advokati, diplomate, čak i određeni prodavci. Govor tela kod tih osoba je veoma bitan za prenos informacije i kao takav, imaju veću svest o svojim pokretima. Ipak i dalje važi da se mikroekspresije veoma teško kontrolišu svojom voljom.

Kao takve, mikroekspresije mogu otkriti skrivene namere i potisnute emocije i tako pomoći psiholozima pri lečenju svojih pacijenata, kao što je pokazano još 1966. godine. Takođe, carinski službenici, obučeni za detekciju mikroekspresija ili opremljeni alatom koji detektuje mikroekspresije, mogu značajno suziti broj ljudi koje treba ispitati i vozila koja treba proveriti. Slična primena se može naći i u policiji.

Mikroekspresije su korisne i pri velikim poslovnim pregovorima, analizi političkih govora, pri ispitivanju osumnjičenih osoba i forenzici, otkrivajući da li su informacije tačne i potpune.

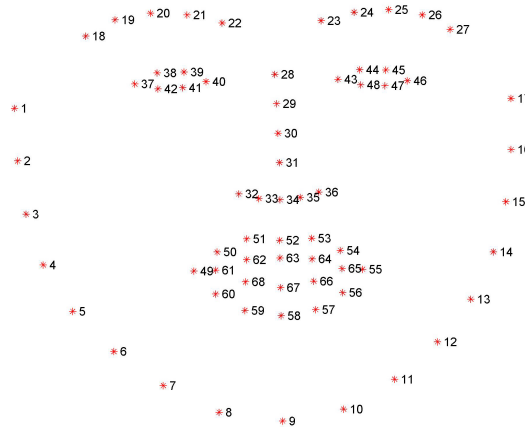
¹U vreme pisanja ovog rada autoru nije poznat nijedan rad koji ukazuje da određene osobe mogu svesno da potisnu mikroekspresije u situacijama od velikog značaja.

Kako su primene mikroekspresija raznovrsne, veliki broj ljudi prolazi kroz obuku uočavanja i prepoznavanja mikroekspresija. Takva obuka je nužno spora i kao takva skupa.² Automatizacija detekcije mikroekspresija bi omogućila da indikatore emocija u vidu mikroekspresija pronalazi računar, čime bi se ubrzao proces obuke. Dodatno to bi omogućilo da proces uočavanja mikroekspresija postane pristupačniji. Ipak postojeći sistemi nisu dovoljno dobri kako bi zamenili ljude u potpunosti u procesu detekcije mikroekspresija.

1.2 Koraci implementacije alata za detekciju mikroekspresija

Pri konstrukciji alata za detekciju mikroekspresija vodićemo se datom definicijom mikroekspresije. Takav alat, stoga, mora registrovati brze i kratke pokrete lica.

Kako je oblast interesovanja lice, neophodno nam je da detektujemo lice osobe kao i ključne tačke lica. Ključne tačke lica su tačke koje opisuju određene karakteristike lica kao što su oblik nosa, lokacije uglova oka, konture obrva i usana. Primer ključnih tačaka lica je dat na [Slici 1.1](#).



Slika 1.1: Primer ključnih tačaka lica.

Ključne tačke lica su neophodne, kako za centriranje lica, tako i da bi se sa sigurnošću utvrdilo da se pomeraji koje detektujemo dešavaju na licu, a ne da su pomeraji glave. Ključne tačke nam takođe omogućuju da promenimo veličinu lica zadržavajući odnose tačaka na licu. Dodatno, lokalizacija mikroekspresija je značajno olakšana upotrebom ključnih tačaka.

Još jedan zahtev koji alat mora da ispuni a koji sledi direktno iz definicije jeste detekcija brzih i kratkih pokreta. Problem koji se nameće pri tom zahtevu jeste da će i treptaji oka biti svrstani kao mikroekspresija. Dati problem se ne može jednostavno otkloniti jer se maksimalni intenzitet mikroekspresija vremenski preklapa sa treptajima oka u oko 50% slučajeva.^{3 4} Uklanjanje regiona oka, pri detekciji mikroekspresija, bi stoga dovelo do smanjenja broja netačno klasifikovanih instanci ali i do smanjenja broja tačno klasifikovanih instanci.

Sličan rezultat bio bi postignut i kada bi se koristio detektor treptaja oka gde bi bile uklonjene sve detektovane instance mikroekspresija za koje je ustanovljeno da

²Paul Ekman koji važi za ljudski detektor laži iscrpno se bavio govorom tela, mikroekspresijama i makroekspresijama preko 40 godina.

³Pri tome se misli na primere iz SMIC baze podataka mikroekspresija.

⁴Naravno ne važi obrnuto: Da je svaki drugi treptaj oka mikroekspresija.

se vremenski poklapaju sa treptajem oka [18]. Dodatno treptaji i pokreti oka mogu nositi informacije o određenim emocijama kao i o nivou stresa osobe.

Iz navedenih razloga detektor će obuhvatiti i detekciju treptaja (kao i brzih pomeraja) oka, dok je za tačnije rezultate neophodno konstruirati dodatni sistem koji bi, na osnovu treptaja i pomeraja oka, utvrdio nivo stresa osobe kao i tekuće emocije.

Iako smo postavili uslov da se registruju brzi pokreti, još uvek nije objašnjen način na koji će to biti urađeno. Ukratko, posmatraćemo niz fiksne veličine koji sadrži uzastopne slike iz video zapisa.⁵ Od početne i krajnje slike niza ćemo formirati "očekivanu sliku" koja bi trebalo da bude što bliža slici iz sredine niza. Ako slika iz sredine niza puno odstupa od očekivane slike smatramo da smo registrovali mikroekspresiju i da je slika deo te mikroekspresije. Detaljniji opis postupka biće dat u [glavi 2](#).

Ako bi konstruisali alat vodeći se samo prethodnim upustvima, naš alat bi imao jako loše rezultate nad video zapisima sa promenljivim osvetljenjem. Iz tog razloga umesto sa slikama video zapisa radićemo sa LBP atributima tih slika. Vrednost LBP atributa ne zavisi od promene osvetljenja. LBP atributi biće opisani u [glavi 2](#).

⁵Broj slika u nizu će zavisi od broja slika koji se prikaže u jednoj sekundi video zapisa

Glava 2

Implementacione tehnike

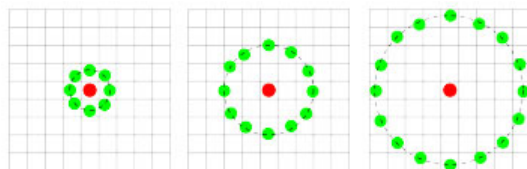
U ovom poglavlju će, po sekcijama, biti prikazane neophodne tehnike za implementaciju automatske detekcije mikroekspresija.

2.1 LBP Atributi

LBP atributi (skraćeno od Local Binary Pattern) su atributi koji služe za opis osnovnih osobina slika kao što su oblik, boja, tekstura i pokret. LBP atributi su veoma zastupljeni u analizi dvodimenzionih tekstura.

LBP atributi opisuju lokalnu strukturu slike tako što porede svaki piksel sa njegovom okolinom. Primer vizualizovanih LBP atributa dat je na slici 2.3.

LBP atributi su definisani sa dva parametra: *Poluprečnikom* (r) i *brojem suseda* (n). Uzimamo jedan po jedan piksel, posmatramo ga kao centralni i poredimo njegov intenzitet u nijansama sive sa okolnih n piksela koji se nalaze na krugu poluprečnika r . Ako je intenzitet centralnog piksela veći od intenziteta okolnog piksela uzimamo 0, inače uzimamo 1. Na taj način formiramo binarni broj dužine n i taj broj zapisujemo u matrici LBP atributa na mestu centralnog piksela.¹ Vizualizacija prethodno opisanog postupka data je slikom 2.1. Vrednosti susednih piksela, obojenih zelenom bojom, se porede sa vrednošću centralnog piksela koji je obojen crvenom bojom.



Slika 2.1: LBP atributi.

Pozicija i -tog susednog piksela, u odnosu na centralni, data je formulama:

$$x = r \cos \frac{2\pi i}{n} \quad (2.1)$$

$$y = -r \sin \frac{2\pi i}{n} \quad (2.2)$$

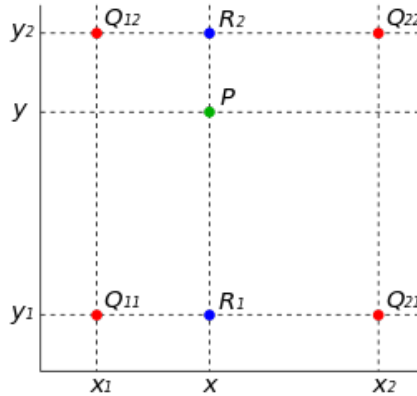
Jasno, matrica LBP atributa je manjih dimenzija nego originalna slika, kako bi za svaki centralni piksel postojali svi okolni pikseli.²

Ovde je bitno napomenuti da iz formula 2.1-2.2 sledi da x i y koordinate piksela, za koje računamo intenzitete u nijansama sive, ne moraju biti celi brojevi. Kako

¹Upisana vrednost je vrednost LBP atributa centralnog piksela

²Ako je slika dimenzije $k \times m$ tada je LBP matrica dimenzije $(k - 2r) \times (m - 2r)$.

vrednostima piksela slike pristupamo isključivo pomoću celobrojnih koordinata neophodno je da intenzitet piksela u nijansama sive izračunamo pomoću intenziteta okolnih piksela sa celobrojnim koordinatama, koristeći neku vrstu interpolacije. Kod LBP atributa uglavnom se koristi bilinearna interpolacija.



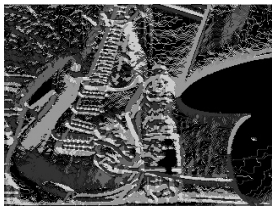
Slika 2.2: Bilinearna interpolacija.

$$f(x, y_1) = \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad (2.3)$$

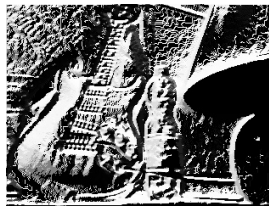
$$f(x, y_2) = \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (2.4)$$

$$f(x, y) = \frac{y_2 - y_1}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \quad (2.5)$$

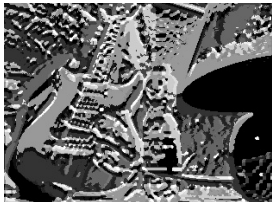
Neka piksel P , prikazan na [slici 2.2](#), ima koordinate (x, y) i neka je njegov intenzitet u nijansama sive dat sa $f(x, y)$. Vrednost $f(x, y)$, u slučaju da x ili y nisu celobrojne vrednosti, nalazimo bilinearnom interpolacijom koja je data formulama 2.3-2.5. $Q_{11}, Q_{12}, Q_{21}, Q_{22}$ predstavljaju piksele sa koordinatama, redom: $(\lfloor x \rfloor, \lfloor y \rfloor)$, $(\lfloor x \rfloor, \lceil y \rceil)$, $(\lceil x \rceil, \lfloor y \rfloor)$, $(\lceil x \rceil, \lceil y \rceil)$.



(A) Slika LBP atributa sa poluprečnikom 1 i 4 suseda.



(B) Slika LBP atributa sa poluprečnikom 1 i 16 suseda.



(C) Slika LBP atributa sa poluprečnikom 4 i 4 suseda.



(D) Slika LBP atributa sa poluprečnikom 4 i 16 suseda.



(E) Originalna slika

Slika 2.3: Primer LBP atributa.

2.2 Detekcija i praćenje ključnih tačaka

Pretpostavimo da imamo trening skup slika lica na kojima su označene lokacije svih n ključnih tačkaka. Lokacije ključnih tačkaka jednog lica možemo opisati vektorom:

$$X = (x_0, y_0, \dots, x_{n-1}, y_{n-1})$$

gde je (x_k, y_k) pozicija k -te ključne tačke na slici. Vektor X ćemo nazivati još i vektorom modela raspodele ključnih tačkaka datog lica i označićemo ga sa PDM (eng. Point Distribution Model) [8]. Svaki PDM vektor se nalazi u prostoru sa $2n$ dimenzija. Svakako je moguće smanjiti dimenzionalnost jer nisu sve koordinate PDM vektora nezavisne.³

Pre nego što pređemo na smanjivanje dimenzionalnosti prvo moramo da izvršimo poravnanje instanci iz trening skupa kako bi lakše mogli da uočimo zavisnosti između koordinata ključnih tačkaka.

2.2.1 Poravnati PDM

Poravnanje vršimo tako što transliramo, rotiramo i skaliramo instance trening skupa. Translaciju vršimo tako što od svakog PDM vektora $\{X\}_{m=1}^S$ oduzmemo njegov prosek:

$$\forall i \quad \hat{x}_i = x_i - \bar{x} \quad , \quad \hat{y}_i = y_i - \bar{y} \quad (2.6)$$

zatim iz trening skupa izaberemo lice po izboru prema kojem ćemo rotirati i skalirati sva ostala lica. Neka je njegov PDM vektor z_1 . Normiramo z_1 tako da važi $\|z_1\| = 1$.

Ostala lica skaliramo i rotiramo tako da se što bolje poravnaju sa z_1 . Ako je PDM vektor j -tog lica trening skupa z_j tada z_j skaliramo za s_j i rotiramo za ugao θ_j .

$$s_j = \sqrt{a_j^2 + b_j^2} \quad \text{i} \quad \theta_j = \tan^{-1} \frac{a_j}{b_j} \quad (2.7)$$

gde su a_j i b_j definisani kao:

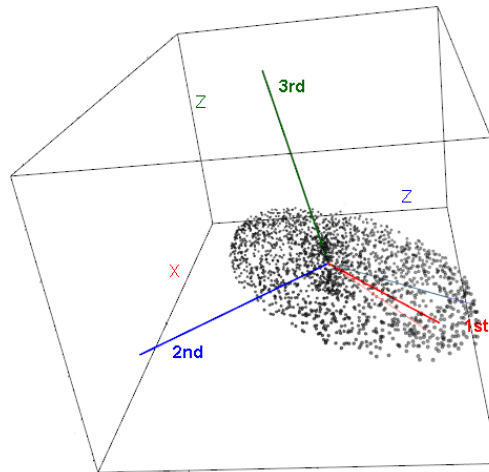
$$a_j = \frac{z_1 \cdot z_j}{\|z_j\|^2} \quad \text{i} \quad b_j = \frac{\sum_{i=1}^n z_{ji}^{(x)} z_{1i}^{(y)} - z_{1i}^{(x)} z_{ji}^{(y)}}{\|z_j\|^2} \quad (2.8)$$

Poravnanje PDM vektora z_j i z_1 se zapravo svodi na nalaženje linearne transformacije (u našem slučaju skaliranja i rotacije) koja podešava ključne tačke iz z_j tako da se što bolje "poklope" sa ključnim tačkama iz z_1 . Kako bi izmerili koliko jedan skup tačkaka odudara od drugog skupa tačkaka koristićemo sumu kvadrata rastojanja između odgovarajućih tačkaka kao meru odudaranja jednog skupa tačkaka od drugog. Nalaženje odgovarajuće linearne transformacije se tada svodi na problem nalaženja parametara s_j i θ_j za koje važi da je suma kvadrata rastojanja minimalna, a čija rešenja su data formulama 2.7-2.8.

³Primer zavisnosti koordinata: koordinate vrha nosa su obavezno između koordinata očiju.

2.2.2 Smanjivanje dimenzionalnosti

Koristićemo analizu glavnih komponenti⁴ kako bi smanjili dimenzionalnost podataka. Analiza glavnih komponenti nam pronalazi pravce duž kojih podaci najviše variraju, kao što je prikazano na slici 2.4.



Slika 2.4: Analiza glavnih komponenti.

Prvi korak je računanje proseka PDM vektora. Ako je S broj lica u trening skupu tada se prosečan PDM vektor μ računa kao:

$$\mu = \frac{1}{S} \sum_{i=1}^S z_i \quad (2.9)$$

Zatim računamo matricu kovarijance Σ

$$\Sigma = \frac{1}{S-1} \sum_{i=1}^S (z_i - \mu)(z_i - \mu)^T \quad (2.10)$$

Neka su sopstvene vrednosti Σ poređane u opadajućem redosledu i označene redom sa $\lambda_1, \dots, \lambda_S$ i neka su njima odgovarajući sopstveni vektori v_1, \dots, v_S . Sopstvene vektore matrice Σ nazivamo još i glavnim komponentama.

Ako želimo da zadržimo bar $K\%$ varijabilnosti podataka, u vidu varijanse, neophodno je da važi:

$$K \geq 100 \frac{\sum_{i=1}^K \lambda_i}{\sum_{j=1}^S \lambda_j}$$

Ako matrica Φ sadrži redom, po kolonama, prvih K glavnih komponenti tada PDM vektor z možemo predstaviti kao

$$\begin{aligned} z &= \mu + \Phi q \\ q &= \Phi^T (z - \mu) \end{aligned} \quad (2.11)$$

Prethodna jednakost važi jer je matrica Φ ortogonalna.

⁴PCA - Principal component analysis

2.2.3 Pronalaženje ključnih tačaka regularizovanim prosečnim pomerajima

Neka je data slika \mathcal{I} koja sadrži lice. Lokaciju i -te ključne tačke na slici \mathcal{I} možemo predstaviti kao:

$$\mathbf{x}_i = sR(\bar{\mathbf{x}}_i + \Phi_i \mathbf{q}) + \mathbf{t} \quad (2.12)$$

gde je s faktor skaliranja, R matrica rotacije, \mathbf{t} vektor translacije i $\bar{\mathbf{x}}_i$ prosečna lokacija i -te ključne tačke poravnatih PDM vektora trening skupa.

Φ_i predstavlja podmatricu vrsta matrice Φ koje se odnose na i -tu ključnu tačku, dok \mathbf{q} prestavlja vektor koordinata u koordinatnom sistemu koje obrazuju glavne komponente dobijene analizom glavnih komponenti.

U jednačini 2.12 prvo nalazimo odgovarajuću lokaciju i -te ključne tačke nad poravnatim PDM vektorom, a zatim sve ključne tačke skaliramo, rotiramo i transliramo tako da se novodobijene lokacije ključnih tačaka PDM vektora što bolje poklapaju sa lokacijama ključnih tačaka na slici \mathcal{I} .⁵

Nazovimo sa $\mathbf{p} = \{s, R, \mathbf{t}, \mathbf{q}\}$ skup svih parametara PDM vektora. Tada problem nalaženja ključnih tačaka lica možemo predstaviti kao problem minimizacije funkcije \mathcal{Q} po \mathbf{p} :

$$\mathcal{Q}(\mathbf{p}) = \mathcal{R}(\mathbf{p}) + \sum_{i=1}^n \mathcal{D}_i(\mathbf{x}_i, \mathcal{I}) \quad (2.13)$$

gde je \mathcal{R} regularizaciona funkcija koja daje velike vrednosti za parametre \mathbf{p} koji obrazuju PDM vektor lica koje je malo verovatno.⁶ \mathcal{D}_i je funkcija koja meri odudaranje lokacije \mathbf{x}_i od prave lokacije i -te ključne tače na slici \mathcal{I} .

Problem 2.13 se može predstaviti i kao maksimizacija verovatnoće parametara PDM vektora pod pretpostavkom da se sve tačke PDM vektora poklapaju sa ključnim tačkama na slici \mathcal{I} . Pri tom pretpostavljamo da verovatnoća poklapanja i -te ključne tačke lica sa slike \mathcal{I} zavisi isključivo od i -te tačke PDM vektora, \mathbf{x}_i , a ne i od okolnih ključnih tačaka.

$$p(\mathbf{p} | \{\ell_i = 1\}_{i=1}^n, \mathcal{I}) \propto p(\mathbf{p}) \prod_{i=1}^n p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I}) \quad (2.14)$$

Jednačina 2.14 sledi direktno, primenom Bajesove teoreme.

$$\begin{aligned} p(\mathbf{p} | \{\ell_i = 1\}_{i=1}^n, \mathcal{I}) &= \frac{p(\mathbf{p}) p(\{\ell_i = 1\}_{i=1}^n | \mathbf{p}, \mathcal{I})}{p(\{\ell_i = 1\}_{i=1}^n | \mathcal{I})} = \frac{p(\mathbf{p}) \prod_{i=1}^n p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I})}{\prod_{i=1}^n p(\ell_i = 1 | \mathcal{I})} \\ &\propto p(\mathbf{p}) \prod_{i=1}^n p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I}) \end{aligned} \quad (2.15)$$

Tada je funkcija regularizacije \mathcal{R} predstavljena kao:

$$\mathcal{R}(\mathbf{p}) = -\ln p(\mathbf{p}) \quad (2.16)$$

dok je funkcija mere odudaranja i -te ključne tačke \mathcal{D}_i oblika:

$$\mathcal{D}_i(\mathbf{x}_i, \mathcal{I}) = -\ln p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I}) \quad (2.17)$$

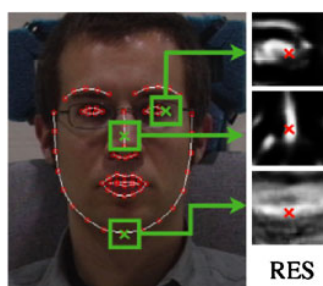
⁵ $\bar{\mathbf{x}}_i + \Phi_i \mathbf{q}$ je zapravo lokacija i -te ključne tačke poravnatog PDM vektora.

⁶ \mathcal{R} će kažnjavati kompleksne oblike lica i zavisi od izbora apriorne raspodele.

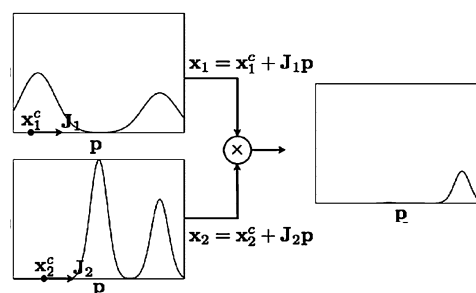
Funkcija koja računa verovatnoću da je i -ta ključna tačka PDM vektora poravnata sa i -tom ključnom tačkom lica slike \mathcal{I} na lokaciji x se zove **ekspert polja** i -te ključne tačke.

$$p(\ell_i = 1 | x, \mathcal{I}) - \text{ekspert polja} \quad (2.18)$$

Eksperti polja su uglavnom lako izračunljive funkcije, koje nisu unapred date i koje je potrebno modelovati. Veoma često su to funkcije koje se određuju metodama mašinskog učenja⁷ i same za sebe nisu veoma precizne. Način određivanja eksperta polja biće naknadno objašnjen. Primeri izračunatih vrednosti eksperata polja (definisanog u [26]) za lokacije piksela iz kvadratne oblasti oko procenjenih lokacija ključnih tačaka su prikazani na slici 2.5A.



(A) Primeri izračunatih vrednosti $p(\ell_i = 1 | x, \mathcal{I})$. Pikseli sa svetlijom bojom odgovaraju većoj verovatnoći da je ključna tačka poravnata na toj lokaciji.



(B) Proizvod izlaznih vrednosti eksperata polja. Na y osi su date vrednosti $p(\ell_i = 1 | x, \mathcal{I})$ sa leve strane (gde $i = 1, 2$) i $p(\ell_i = 1 | x, \mathcal{I})p(\ell_2 = 1 | x, \mathcal{I})$ sa desne strane slike.

Slika 2.5: Eksperti polja.

Postavlja se pitanje: zašto koristimo neprecizne, a lako izračunljive funkcije za eksperte polja? Iako su pojedinačni eksperti polja neprecizni, proizvod njihovih izlaznih vrednosti odstranjuje veliki broj neodređenosti za izbor parametra p , a time i za lokacije tačaka $\{x_i\}_{i=1}^n$, što možemo videti na slici 2.5B.

Eksperti polja uglavnom neće biti glatke funkcije. Takođe poseduju neodređenost jer će svaki ekspert polja posmatrati malu oblast oko lokacije x za koju je potrebno odrediti verovatnoću poravnanja ključne tačke. Iz tih razloga, kao i činjenice da su postojeće metode optimizacija veoma spore za eksperte polja u trenutnom obliku⁸, neophodno je aproksimirati eksperte polja jednostavnijim, parametrizovanim i glatkim funkcijama. Optimizaciju ćemo nakon toga vršiti nad aproksimacijama eksperta polja.

⁷Te funkcije se "uče"

⁸Iako nismo dali konkretan oblik eksperta polja, ovde se misli generalno na sve funkcije koje se uče, a koje bi bile dobri kandidati za eksperte polja. Za konkretan oblik eksperta polja pogledati [26]

2.2.3.1 Aproksimacija gausovim raspodelama

Ekspert polja i -te ključne tačke aproksimiramo kao:

$$p(\ell_i | \mathbf{x}_i, \mathcal{I}) \approx \sum_{k=1}^{K_i} \pi_{ik} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (2.19)$$

gde je K_i broj gausovih raspodela kojim aproksimiramo ekspert polja i -te ključne tačke, a π_{ik} su koeficijenti za koje važi $\sum_{k=1}^{K_i} \pi_{ik} = 1$. Koristeći EM algoritam [5] za ekspert polja i -te ključne tačke nalazimo parametre $\{\pi\}_{ik}$, $\{\boldsymbol{\Sigma}\}_{ik}$ i $\{\boldsymbol{\mu}\}_{ik}$. Pri tom smo skup podataka, za potrebe primene EM algoritma, formirali koristeći prave eksperte polja.

Ako je z_i ⁹ latentna promenljiva za koju važi

$$\sum_{k=1}^{K_i} z_{ik} = 1 \quad , \quad z_{ik} \in \{0, 1\}$$

zajedničku raspodelu promenljivih z_{ik} i ℓ_i možemo predstaviti kao:

$$p(\ell_i, z_{ik} = 1 | \mathbf{x}_i, \mathcal{I}) = p(z_{ik} = 1 | \mathbf{x}_i, \mathcal{I}) p(\ell_i | z_{ik} = 1, \mathbf{x}_i, \mathcal{I}) \quad (2.20)$$

gde važi:

$$p(z_{ik} = 1 | \mathbf{x}_i, \mathcal{I}) = \pi_{ik} \quad \wedge \quad p(\ell_i | z_{ik} = 1, \mathbf{x}_i, \mathcal{I}) = \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (2.21)$$

Jednačina 2.14 tada ima oblik:

$$p(p) \prod_{i=1}^n \sum_{k=1}^{K_i} p(z_{ik} = 1, \ell_i = 1 | \mathbf{x}_i, \mathcal{I})^{10} \quad (2.22)$$

Kako je teško optimizovati funkciju u datom obliku posmatrajmo zajedničku raspodelu latentne promenljive z_i i promenljive ℓ_i . Iz formula 2.21 znamo da važi:

$$p(z_i | \mathbf{x}_i, \mathcal{I}) = \prod_{k=1}^{K_i} \pi_{ik}^{z_{ik}} \quad \wedge \quad p(\ell_i | z_i, \mathbf{x}_i, \mathcal{I}) = \prod_{k=1}^{K_i} \mathcal{N}^{z_{ik}}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}) \quad (2.23)$$

Zajednička raspodela je tada oblika:

$$p(\{\ell_i\}_{i=1}^n, \mathbf{z}_i | \mathbf{p}, \mathcal{I}) = \prod_{i=1}^n p(\ell_i, z_i | \mathbf{x}_i, \mathcal{I}) = \prod_{i=1}^n \prod_{k=1}^{K_i} (\pi_{ik} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}))^{z_{ik}} \quad (2.24)$$

Maksimizacija 2.24 se svodi na maksimizaciju:

$$\ln p(\{\ell_i\}_{i=1}^n, \mathbf{z}_i | \mathbf{p}, \mathcal{I}) = \sum_{i=1}^n \sum_{k=1}^{K_i} z_{ik} (\ln \pi_{ik} + \ln \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})) \quad (2.25)$$

Kako su vrednosti latentnih promenljivih z_i nepoznate, u prethodnoj formuli ćemo umesto z_{ik} koristiti njeno očekivanje. Stoga ćemo, primenom EM algoritma,

⁹ z_{ik} će biti 1 u slučaju da je k -ta gausova raspodela generisala uzorak \mathbf{x}_i

¹⁰Problem kod ove jednačine to je što logaritam ne može da prođe kroz sumu kao kroz proizvod, pa bi direktna optimizacija ove formule bila teška

maksimizovati 2.14. Pri tom je, u svakom koraku EM algoritma, potrebno da izračunamo:

$$E[\mathbf{z}_{ik}] = p(\mathbf{z}_{ik} = 1 | \ell_i, \mathbf{x}_i, \mathcal{I}) = \frac{\pi_{ik} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})}{\sum_{j=1}^{K_i} \pi_{ij} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ij}, \boldsymbol{\Sigma}_{ij})} \quad (2.26)$$

Zatim pristupamo koraku maksimizacije 2.25, što je isto kao i minimizacija sledećeg očekivanja:¹¹

$$\mathcal{Q}(\mathbf{p}) = E_z[-\ln p(\mathbf{p}) \prod_{i=1}^n p(\ell_i = 1, \mathbf{z}_i | \mathbf{x}_i, \mathcal{I})] \quad (2.27)$$

Stav 2.2.1 Neka je $p(\mathbf{p}) = \mathcal{N}(\mathbf{q} | \mathbf{0}, \Lambda)$, gde je

$$\Lambda_{ij} = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases}$$

Pri čemu su λ_i sopstvene vrednosti dobijene analizom glavnih komponenti.¹² Minimizacija očekivanja 2.27 po parametru \mathbf{p} može se svesti na minimizaciju formule:

$$\|\mathbf{q}\|_{\Lambda^{-1}}^2 + \sum_{i=1}^n \sum_{k=1}^{K_i} E[\mathbf{z}_{ik}] \|\mathbf{x}_i - \boldsymbol{\mu}_{ik}\|_{\boldsymbol{\Sigma}_{ik}^{-1}}^2 \quad (2.28)$$

Dokaz:

$$\begin{aligned} E_z[-\ln p(\mathbf{p}) \prod_{i=1}^n p(\ell_i = 1, \mathbf{z}_i | \mathbf{x}_i, \mathcal{I})] &= E_z[-\ln p(\mathbf{p}) \prod_{i=1}^n \prod_{k=1}^{K_i} (\pi_{ik} \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik}))^{\mathbf{z}_{ik}}] = \\ &= -\ln \frac{1}{(2\pi)^{\frac{m}{2}} |\Lambda|^{\frac{1}{2}}} e^{-\frac{1}{2} \mathbf{q}^T \Lambda^{-1} \mathbf{q}} + \sum_{i=1}^n \sum_{k=1}^{K_i} E[\mathbf{z}_{ik}] (-\ln \pi_{ik} - \ln \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_{ik}, \boldsymbol{\Sigma}_{ik})) \propto \\ &= \frac{1}{2} \|\mathbf{q}\|_{\Lambda^{-1}}^2 + \sum_{i=1}^n \sum_{k=1}^{K_i} -E[\mathbf{z}_{ik}] \ln \frac{1}{(2\pi)^{\frac{D}{2}} \boldsymbol{\Sigma}_{ik}^{\frac{1}{2}}} e^{-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_{ik})^T \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_{ik})} = \\ &\propto \|\mathbf{q}\|_{\Lambda^{-1}}^2 + \sum_{i=1}^n \sum_{k=1}^{K_i} E[\mathbf{z}_{ik}] \|\mathbf{x}_i - \boldsymbol{\mu}_{ik}\|_{\boldsymbol{\Sigma}_{ik}^{-1}}^2 \blacksquare \end{aligned}$$

Minimizacija 2.28 se postiže iterativno, Gaus-Njutnovom¹³ metodom optimizacije [22] sa Tihonevljevom regularizacijom. Korak $\Delta \mathbf{p}$ se tada dobija na sledeći način:

$$\Delta \mathbf{p} = -H_{GMM}^{-1} (\tilde{\Lambda}^{-1} \mathbf{p} + \sum_{i=1}^n \sum_{k=1}^{K_i} E[\mathbf{z}_{ik}] \mathbf{J}_i^T \boldsymbol{\Sigma}_{ik}^{-1} (\mathbf{x}_i^{(c)} - \boldsymbol{\mu}_{ik})) \quad (2.29)$$

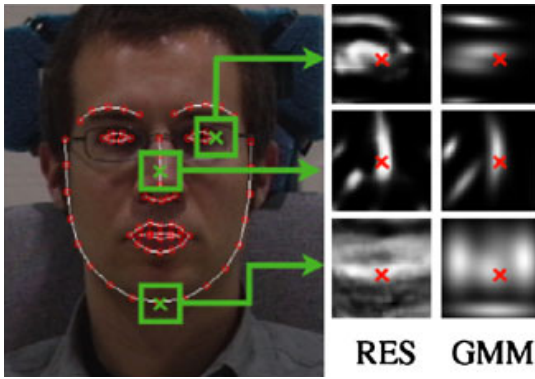
gde je $\mathbf{x}_i(\mathbf{p}_n) = \mathbf{x}_i^{(c)}$ trenutna procena i -te ključne tačke PDM vektora, a matrica H_{GMM} aproksimacija matrice hesijana sa regularizacijom.

$$H_{GMM} = \tilde{\Lambda}^{-1} + \sum_{i=1}^n \sum_{k=1}^{K_i} E[\mathbf{z}_{ik}] \mathbf{J}_i^T \boldsymbol{\Sigma}_{ik}^{-1} \mathbf{J}_i \quad (2.30)$$

¹¹Jedina razlika je što umesto korišćenja promenljive \mathbf{z}_{ik} , što nemamo, koristimo njeno očekivanje, koje znamo da izračunamo

¹²Vrednosti λ_i odgovaraju glavnim komponentama koje formiraju prostor koji sadrži unapred izabrani procenat varijabilnosti

¹³Gaus-Njutnova metoda je kvazi-Njutnova metoda koja umesto matrice hesijana koristi njenu aproksimaciju.



Slika 2.6: Rezultat aproksimacija eksperata polja određenih ključnih tačaka, gausovim raspodelama.

(RES) Rezultat pravih eksperata polja, $p(\ell_i = 1|x, \mathcal{I})$ za odgovarajuće kvadratne regione oko ključnih tačaka.

(GMM) Rezultat aproksimacije pravih eksperata polja gausovim raspodelama 2.19 gde je $K_i = 5$.

$J = [J_1, \dots, J_n]$ je jakobijan parametara PDM vektora koji smo dobili primenom Tejlorovog razvoja prvog reda na funkcije $x_i(\mathbf{p}_{n+1})$ u tački \mathbf{p}_n :

$$x_i(\mathbf{p}_{n+1}) \approx x_i(\mathbf{p}_n) + \sum_{j=1}^r \frac{\partial x_i(\mathbf{p}_n)}{\partial p_j} \Delta p_j \quad (2.31)$$

gde je r broj elemenata parametara PDM vektora \mathbf{p} . Ako je m broj elemenata vektora \mathbf{q} , tada je matrica $\tilde{\Lambda}^{-1}$ definisana kao:

$$\tilde{\Lambda}_{ij}^{-1} = \begin{cases} 0 & i = j, \quad i \leq (r - m) \\ \frac{1}{\lambda_i} & i = j, \quad i > (r - m) \\ 0 & i \neq j \end{cases}$$

Rezultat rada aproksimiranih eksperata polja se može videti na Slici 2.6. Aproksimacija, iako ne savršena, opisuje funkciju pravog eksperta polja veoma dobro. Ipak postojeća aproksimacije eksperata polja ima ozbiljne mane.

Pri radu sa mikroekspresijama poželjan je video zapis koji sadrži puno slika u jednoj sekundi. Kako nalaženje ključnih tačaka, koristeći aproksimaciju 2.19, traje okvirno 66 sekundi za 1 sekund video zapisa¹⁴, zaključujemo da moramo popraviti postojeću aproksimaciju kako bi vreme izvršavanja našeg programa bilo razumno.

Dakle problem aproksimacije 2.19 pravog eksperta polja je vremenska složenost optimizacije. Koren ovog problema potiče iz primene EM algoritma za eksperte polja svih ključnih tačaka, što je vremenski veoma zahtevno. Još jedan značajan problem aproksimacije 2.19 jeste nestabilnost. Kako Gaus-Njutnova metoda konvergira lokalno, veliki broj gausovih raspodela koje učestvuju u aproksimaciji mogu rezultovati završetkom optimizacije u lokalnom optimumu.

2.2.3.2 Regularizovani prosečni pomeraj

Pri primeni analize glavnih komponenata deo sopstvenih vektora matrice kovarijance je odbačen kako bi smanjili dimenzionalnost problema. Iz tog razloga računski nećemo moći rekonstruisati lokacije ključnih tačaka savršeno, ali ako smo zadržali dovoljan procenat varijabilnosti možemo smatrati da greška procene lokacija potiče od šuma podataka. Taj šum možemo modelovati Gausovom raspodelom.

$$\mathbf{y}_i = \mathbf{x}_i + \boldsymbol{\epsilon}_i \quad \text{gde važi} \quad \boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{\epsilon}_i | \mathbf{0}, \rho \mathbf{I}) \quad (2.32)$$

¹⁴Ovde se pri tom misli na video zapis koji sadrži 30 slika u sekundi. Pri radu sa mikroekspresijama nije retkost da video zapisi sadrže i do 200 slika u sekundi

Moghadam i Pentland su dokazali da se varijansa šuma ρ može izračunati iz trening skupa [20].

$$\rho = \frac{1}{N - m} \sum_{i=m+1}^N \lambda_i \quad (2.33)$$

Iz 2.33 može se izvesti zaključak da ρ predstavlja aritmetičku sredinu svih sopstvenih vrednosti koje su dobijene analizom glavnih komponenti i čiji se odgovarajući sopstveni vektori ne nalaze u Φ .

Stav 2.2.2 Neka je Ψ_i skup kandidata za lokaciju i -te ključne tačke. Tada se verovatnoća poravnanja na pravoj lokaciji ključne tačke, \mathbf{x}_i , može izračunati kao:

$$p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I}) = \sum_{\mathbf{y}_i \in \Psi_i} p(\ell_i = 1 | \mathbf{y}_i, \mathcal{I}) p(\mathbf{y}_i | \mathbf{x}_i, \mathcal{I}) \quad (2.34)$$

Dokaz:

$$p(\ell_i = 1 | \mathbf{x}_i, \mathcal{I}) = \sum_{\mathbf{y}_i \in \Psi_i} p(\ell_i = 1, \mathbf{y}_i | \mathbf{x}_i, \mathcal{I}) = \sum_{\mathbf{y}_i \in \Psi_i} p(\ell_i = 1 | \mathbf{y}_i, \mathbf{x}_i, \mathcal{I}) p(\mathbf{y}_i | \mathbf{x}_i, \mathcal{I})$$

gde važi

$$p(\ell_i = 1 | \mathbf{y}_i, \mathbf{x}_i, \mathcal{I}) = \frac{p(\ell_i = 1, \mathbf{y}_i | \mathbf{x}_i, \mathcal{I})}{p(\mathbf{y}_i | \mathbf{x}_i, \mathcal{I})} = \frac{p(\ell_i = 1, \mathbf{x}_i | \mathbf{y}_i, \mathcal{I})}{p(\mathbf{x}_i | \mathbf{y}_i, \mathcal{I})} \quad (2.35)$$

Dodatno, na osnovu 2.32 sledi

$$p(\mathbf{y}_i | \mathbf{x}_i, \mathcal{I}) = \mathcal{N}(\mathbf{y}_i | \mathbf{x}_i, \rho \mathbf{I}) = \mathcal{N}(\mathbf{x}_i | \mathbf{y}_i, \rho \mathbf{I}) = p(\mathbf{x}_i | \mathbf{y}_i, \mathcal{I}) \quad (2.36)$$

Iz 2.35 i 2.36 zaključujemo

$$p(\mathbf{x}_i | \ell_i = 1, \mathbf{y}_i, \mathcal{I}) p(\ell_i = 1 | \mathbf{y}_i, \mathcal{I}) = p(\ell_i = 1, \mathbf{x}_i | \mathbf{y}_i, \mathcal{I}) = p(\ell_i = 1, \mathbf{y}_i | \mathbf{x}_i, \mathcal{I}) \quad (2.37)$$

Verovatnoću da je i -ta ključna tačka poravnata na lokaciji \mathbf{y}_i računamo na osnovu prave funkcije eksperta polja.

$$\pi_{\mathbf{y}_i} = p(\ell_i = 1 | \mathbf{y}_i, \mathcal{I}) \quad (2.38)$$

Odatle iz 2.37 i 2.38 dobijamo

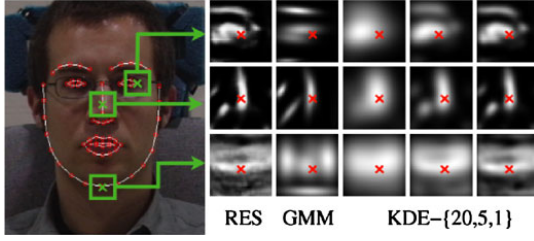
$$\begin{aligned} p(\mathbf{x}_i | \ell_i = 1, \mathbf{y}_i, \mathcal{I}) \pi_{\mathbf{y}_i} &= p(\ell_i = 1, \mathbf{y}_i | \mathbf{x}_i, \mathcal{I}) = \pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i | \mathbf{y}_i, \rho \mathbf{I}) \\ \implies p(\mathbf{x}_i | \ell_i = 1, \mathbf{y}_i, \mathcal{I}) &= \mathcal{N}(\mathbf{x}_i | \mathbf{y}_i, \rho \mathbf{I}) \\ \implies p(\ell_i = 1 | \mathbf{x}_i, \mathbf{y}_i, \mathcal{I}) &= p(\ell_i = 1 | \mathbf{y}_i, \mathcal{I}) \quad \text{Na osnovu svojstva 2.35} \end{aligned}$$

■

Razlika između aproksimacije 2.19 i 2.34 je ta što je aproksimacija data u 2.34 nparametarska procena pravog eksperta polja i kvalitet aproksimacije zavisi od izbora kandidat-lokacija $\{\mathbf{y}_i | \mathbf{y}_i \in \Psi_i\}$.

Slično kao kod aproksimacije gausovim raspodelama, maksimizacija 2.14 se postiže EM algoritmom[5]. U svakoj iteraciji računamo očekivanje

$$E[\mathbf{z}_{iy_i}] = p(\mathbf{y}_i | \ell_i = 1, \mathbf{x}_i, \mathcal{I}) = \frac{\pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i | \mathbf{y}_i, \rho \mathbf{I})}{\sum_{\mathbf{z}_i \in \Psi_i} \pi_{\mathbf{z}_i} \mathcal{N}(\mathbf{x}_i | \mathbf{z}_i, \rho \mathbf{I})} \quad (2.39)$$



Slika 2.7: Rezultat aproksimacije 2.34 eksperata polja određenih ključnih tačaka.

(RES) Rezultat pravih eksperata polja, $p(\ell_i = 1 | \mathbf{x}, \mathcal{I})$ za odgovarajuće kvadratne regione oko ključnih tačaka.

(GMM) Rezultat aproksimacije pravih eksperata polja gausovim raspodelama 2.19 gde je $K_i = 5$.

(KDE) Rezultati aproksimacije 2.34 gde je $\rho = \{20, 5, 1\}$ redom s leva na desno.

Očekivanje, dato prethodnom formulom, dalje koristimo za minimizaciju očekivanja 2.27. Minimizacija očekivanja sledi isto kao iz stava 2.2.1, uzimajući u obzir aproksimaciju eksperata polja datog formulom 2.34.

Minimizacija očekivanja 2.27 se postiže iterativno, Gaus-Njutnovom metodom, gde je korak $\Delta \mathbf{p}$ dat sa:

$$\Delta \mathbf{p} = -H_{KDE}^{-1} (\tilde{\Lambda}^{-1} \mathbf{p} + \sum_{i=1}^n \sum_{\mathbf{y}_i \in \Psi_i} E[z_{i\mathbf{y}_i}] \mathbf{J}_i^T \frac{1}{\rho} (\mathbf{x}_i^{(c)} - \mathbf{y}_i)) \quad (2.40)$$

Aproksimacija matrice hesijana H_{KDE} je data sa:

$$H_{GMM} = \tilde{\Lambda}^{-1} + \sum_{i=1}^n \sum_{\mathbf{y}_i \in \Psi_i} E[z_{i\mathbf{y}_i}] \frac{1}{\rho} \mathbf{J}_i^T \mathbf{J}_i \quad (2.41)$$

Sređivanjem formule 2.40 dobijamo formulu:¹⁵

$$\Delta \mathbf{p} = -(\rho \tilde{\Lambda}^{-1} + \mathbf{J}^T \mathbf{J})^{-1} (\rho \tilde{\Lambda}^{-1} \mathbf{p} - \mathbf{J}^T \mathbf{v}) \quad (2.42)$$

gde se vektor $\mathbf{v} = [v_1, \dots, v_n]$ zove još i vektor prosečnog pomeraja [7].

$$v_i = \left(\sum_{\mathbf{y}_i \in \Psi_i} E[z_{i\mathbf{y}_i}] \mathbf{y}_i \right) - \mathbf{x}_i^{(c)} \quad (2.43)$$

Rezultat rada aproksimiranih eksperata polja koristeći aproksimaciju 2.34 se može videti na Slici 2.7. Ono što treba primetiti jeste da parametar ρ služi kao parametar glatkosti aproksimacije. Što je ρ veće, aproksimacija pravih eksperata polja je lošija ali je funkcija aproksimacije glatkija.¹⁶

Aproksimacije funkcija sa većim ρ mogu da se posmatraju i kao glatkije aproksimacije funkcija sa manjim ρ . Stav Carreira-Perpinan-Williams [6] potvrđuje naš intuitivni zaključak:

Stav 2.2.3 Postoji $\rho < \infty$ tako da je raspodela $\sum_{\mathbf{y}_i \in \Psi_i} \pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i | \mathbf{y}_i, \rho \mathbf{I})$ unimodalna, nezavisno od raspodele uzoraka.

Prethodna teorema nam omogućava da težimo globalnom optimumu umesto lokalnom, što je bio problem pri aproksimaciji gausovim raspodelama. To postizemo tako što nađemo optimalne lokacije ključnih tačaka za aproksimaciju 2.34 sa velikim ρ . Optimalne lokacije iz prethodnog koraka zatim koristimo kao početne lokacije pri optimizaciji čije je rešenje dato sa 2.40, gde se uzima aproksimacija 2.34 sa manjim

¹⁵Formula sledi iz činjenice da je $\sum_{\mathbf{y}_i \in \Psi_i} E[z_{i\mathbf{y}_i}] = 1$

¹⁶Pri tom se ovde ne misli na glatkost u matematičkom smislu već u intuitivnom ("Brdovitije funkcije" su manje glatke). U matematičkom smislu sve aproksimacije date sa 2.34 su glatke.

ρ nego u prethodnom koraku. Na ovaj način smo otklonili sve bitne probleme koje je aproksimacija 2.19 imala i aproksimaciju 2.34 ćemo koristiti kao aproksimaciju pravog eksperta polja pri traženju lokacija ključnih tačaka.

Algoritam 1: Algoritam za nalaženje lokacija ključnih tačaka

```

Data: Slika  $\mathcal{I}$ , Trenutna procena  $\mathbf{p}$ 
Result:  $\mathbf{p}$ 
// Računamo vrednosti eksperta polja za piksele iz kvadratnog
// regiona centriranog u trenutnoj proceni ključne tačke.
for  $l = 1; l < \text{broj ključnih tačaka}; i ++$  do
     $p(\ell_i = 1 | \mathbf{X}) = \Sigma \mathbf{d}$ ;
    // Formula 2.55
end
while  $\mathbf{p}$  ne konvergira do
    Linearizacija funkcije 2.12:  $\mathbf{x}_i \approx \mathbf{x}_i^{(c)} + \mathbf{J}_i \Delta \mathbf{p}$ ;
    // Računamo vektor prosečnog pomeraja
     $\mathbf{v}_i = \sum_{\mathbf{y}_i \in \Psi_i} \left( \frac{\pi_{\mathbf{y}_i} \mathcal{N}(\mathbf{x}_i^{(c)} | \mathbf{y}_i, \rho \mathbf{I})}{\sum_{\mathbf{z}_i \in \Psi_i} \pi_{\mathbf{z}_i} \mathcal{N}(\mathbf{x}_i^{(c)} | \mathbf{z}_i, \rho \mathbf{I})} \mathbf{y}_i \right) - \mathbf{x}_i^{(c)}$ ;
    // Računamo pomeraj parametara  $\mathbf{p}$ 
     $\Delta \mathbf{p} = -(\rho \tilde{\Lambda}^{-1} + \mathbf{J}^T \mathbf{J})^{-1} (\rho \tilde{\Lambda}^{-1} \mathbf{p} - \mathbf{J}^T \mathbf{v})$ ;
     $\mathbf{p} = \mathbf{p} + \Delta \mathbf{p}$ ;
end
return  $\mathbf{p}$ ;

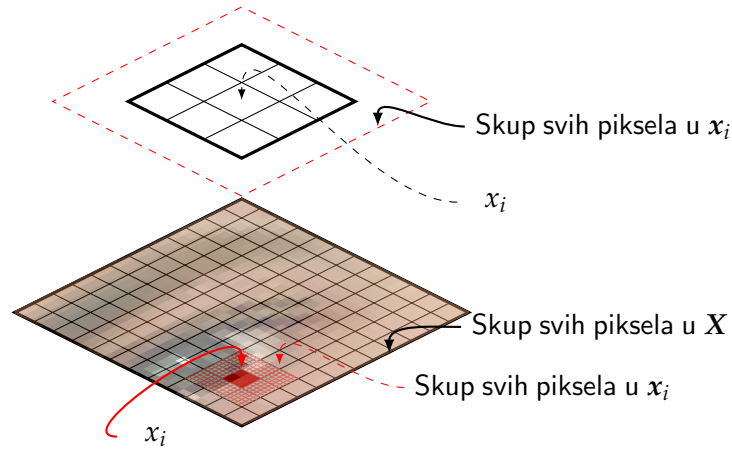
```

2.2.4 Eksperti polja

Iako smo pokazali kako možemo da aproksimiramo eksperte polja funkcijama koje imaju lepa svojstva i dalje nismo pokazali kako funkcija pravog eksperta polja izgleda. Postoje razni oblici za funkciju eksperta polja, a jedan način jeste da se izračuna sigmoidna funkcija logističkog regresora koji je obučen da razlikuje lokacije na kojima se ključna tačka PDM vektora poklapa sa ključnom tačkom na slici od onih gde to ne važi [26].

Ovaj oblik funkcije eksperata polja je previše pojednostavljen. Veze između ulaznih piksela i izlaznih vrednosti eksperta polja su uglavnom složene i nelinearne po prirodi i često postoje veze između susednih piksela.¹⁷ Polje eksperta dato u [26] ne može da se izbori sa prethodnim uslovima i stoga je neophodno konstruisati funkciju koja je lako izračunljiva ali koristi prethodno navedene uslove povezanosti.

Argumenti funkcije eksperta polja su $\mathbf{X} = \{x_1, \dots, x_t\}$, a izlazna vrednost je $\mathbf{y} = \{y_1, \dots, y_t\}$ gde je x_i skup piksela u okolini tačke x_i predstavljen u obliku vektora¹⁸, a y_i verovatnoća da se na lokaciji x_i ključna tačka PDM vektora i slike poklapaju. Vizualna reprezentacija ulaznih parametara se može videti na Slici 2.8.



Slika 2.8: Vizualna reprezentacija ulaznih parametara \mathbf{X} .

Gustinu raspodele \mathbf{y} modelujemo na sledeći način:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{\Psi}}{\int_{-\infty}^{+\infty} e^{\Psi} d\mathbf{y}} \quad (2.44)$$

Rešenje eksperta polja za ulazne parametre \mathbf{X} je tada $\bar{\mathbf{y}}$ za koje važi:

$$\bar{\mathbf{y}} = \arg \max_{\mathbf{y}} p(\mathbf{y}|\mathbf{X}) \quad (2.45)$$

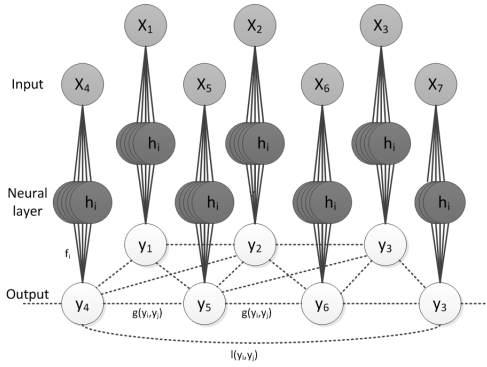
Funkcija Ψ je oblika:

$$\Psi = \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k f_k(y_i, \mathbf{X}, \boldsymbol{\theta}_k) + \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_2} \beta_k g_k(y_i, y_j) + \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_3} \gamma_k l_k(y_i, y_j) \quad (2.46)$$

Parametri modela koje je potrebno naučiti su tada $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_{K_1}\}$, $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_{K_1}\}$, $\boldsymbol{\beta} = \{\beta_1, \dots, \beta_{K_2}\}$ i $\boldsymbol{\gamma} = \{\gamma_1, \dots, \gamma_{K_3}\}$. Gde je $\boldsymbol{\theta}_k \in \mathcal{R}^s$ vektor a $\alpha_k, \beta_k, \gamma_k$ su skalari.

¹⁷Jedna veza je očigledna: bliski pikseli treba da imaju bliske izlazne vrednosti eksperta polja.

¹⁸ x_i je tada element iz \mathcal{R}^s , gde je s broj piksela u okolini x_i koju posmatramo.



Slika 2.9: Prikaz kako je funkcija Ψ konstruisana. Vektori x_i prolaze kroz jedan sloj neuralne mreže pre nego što, u kombinaciji sa ograničenjima koje nameću funkcije g i l , dobijemo izlaznu vrednost y_i .

Formula 2.46 rešava sve probleme koje smo prethodno naveli koristeći funkcije f , g i l . Okolina tačke x_i u obliku vektora x_i direktno doprinosi verovatnoći y_i preko funkcija $\{f_k\}_{k=1}^{K_1}$ koje se definišu kao:

$$f_k(y_i, \mathbf{X}, \boldsymbol{\theta}_k) = -(y_i - h(\boldsymbol{\theta}_k, \mathbf{x}_i))^2$$

$$h(\boldsymbol{\theta}, \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}} \quad (2.47)$$

Funkcije $\{g_k\}_{k=1}^{K_2}$ služe da izlazne vrednosti bliskih piksela budu slične:

$$g_k(y_i, y_j) = -\frac{1}{2} S_{i,j}^{(g_k)} (y_i - y_j)^2 \quad (2.48)$$

$S_{i,j}^{(g_k)}$ je jednaka 1 za bliske piksele a inače je 0. Konkretno, za primene eksperta polja, postavivemo da je $S_{i,j}^{g_1}$ jednako 1 ako su pikseli x_i i x_j susedni a inače 0 i $S_{i,j}^{g_2}$ je jednako 1 za dijagonalne piksele x_i i x_j a inače 0.¹⁹

Funkcije $\{l_k\}_{k=1}^{K_3}$ služe da smanje broj lokalnih maksimuma u skupu izlaznih vrednosti $\{y_1, \dots, y_t\}$:

$$l_k(y_i, y_k) = -\frac{1}{2} S_{i,j}^{l_k} (y_i + y_j)^2 \quad (2.49)$$

Funkcije $S_{i,j}^{l_k}$ služe da se ustanovi koliko su retki lokalni maksimumi i vraća 1 ako su pikseli x_i i x_j na odgovarajućoj udaljenosti a inače 0. Za primene eksperta polja definišemo $S_{i,j}^{l_1}$ jedanko 1 ako su pikseli x_i i x_j udaljeni $r + 3$ ivica, gde je $r = \{1, 2, 3\}$. Grafički prikaz konstrukcije funkcije Ψ možemo videti na Slici 2.9.

2.2.4.1 Učenje eksperta polja

Za učenje eksperta polja neophodno je da imamo skup trening podataka $\{\mathbf{x}^q, \mathbf{y}^q\}_{q=1}^M$ gde su $\mathbf{x}^q = \{x_1^q, \dots, x_t^q\}$ i $\mathbf{y}^q = \{y_1^q, \dots, y_t^q\}$. Ako imamo datu tačnu lokaciju ključne tačke za svaki element trening skupa, $\{z^q\}_{q=1}^M$, verovatnoće y_i^q se tada modeluju sa $y_i^q = \mathcal{N}(x_i^q | z^q, \mathbf{I})$.

Ovde je bitno napomenuti da ekspert polja računa verovatnoću poravnanja jedne ključne tačke, pa ćemo za potrebe traženja svih ključnih tačaka na slici \mathcal{I} morati da istreniramo onoliko eksperata polja koliko imamo ključnih tačaka.

¹⁹Za različite potrebe, pojam bliskosti, možemo definisati na različite načine, ali za potrebe traženja lokacije ključne tačke ovo je sasvim dovoljno.

Želimo da nađemo parametre $\{\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\Theta}\}$ tako da važi:

$$(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\Theta}) = \arg \max_{\alpha, \beta, \gamma, \Theta} \sum_{q=1}^M \ln p(\mathbf{y}^q | \mathbf{x}^q) \quad (2.50)$$

Za rešavanje problema 2.50 zgodno je pokazati sledeću teoremu:

Teorema 2.2.1 Formula 2.44 se može predstaviti u obliku

$$p(\mathbf{y} | \mathbf{X}) = \mathcal{N}(\mathbf{y} | \boldsymbol{\mu}, \Sigma) \quad (2.51)$$

Dokaz: Neka je \mathbf{X} matrica koja u koloni i sadrži vektor \mathbf{x}_i . Funkciju Ψ možemo posmatrati iz delova:

$$\begin{aligned} - \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k (y_i - h(\boldsymbol{\theta}_k, \mathbf{x}_i))^2 &= - \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k y_i^2 + 2 \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k y_i h(\boldsymbol{\theta}_k, \mathbf{x}_i) - \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k h^2(\boldsymbol{\theta}_k, \mathbf{x}_i) \\ &= -\mathbf{y}^T \mathbf{A} \mathbf{y} + \mathbf{y}^T \mathbf{d} - c \end{aligned}$$

gde su

$$A_{ij} = \begin{cases} \sum_{k=1}^{K_1} \alpha_k & i = j \\ 0 & i \neq j \end{cases}, \quad \mathbf{d}^T = 2\boldsymbol{\alpha}^T h(\boldsymbol{\Theta}^T \mathbf{X}) \quad \text{i} \quad c = \sum_{i=1}^t \sum_{k=1}^{K_1} \alpha_k h^2(\boldsymbol{\theta}_k, \mathbf{x}_i)$$

Pri tom $h(\boldsymbol{\Theta}^T \mathbf{X})$ predstavlja primenu sigmoidne funkcije h na elemente matrice $\boldsymbol{\Theta}^T \mathbf{X}$.

Deo funkcije Ψ koji utiče da bliski pikseli imaju slične izlazne vrednosti ima oblik:

$$\begin{aligned} - \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_2} \beta_k \frac{1}{2} S_{i,j}^{(g_k)} (y_i - y_j)^2 &= - \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_2} \beta_k S_{i,j}^{(g_k)} y_i^2 + \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_2} \beta_k S_{i,j}^{(g_k)} y_i y_j \\ &= - \sum_{k=1}^{K_2} \beta_k \sum_{i=1}^t \sum_{j=1}^t y_i^2 S_{i,j}^{(g_k)} + \sum_{k=1}^{K_2} \beta_k \sum_{i=1}^t \sum_{j=1}^t S_{i,j}^{(g_k)} y_i y_j = -\mathbf{y}^T \mathbf{B} \mathbf{y} \end{aligned}$$

Pri tom smo uveli matricu \mathbf{B} oblika:

$$B_{i,j} = \begin{cases} \sum_{k=1}^{K_2} \beta_k \sum_{r=1}^t S_{i,r}^{(g_k)} - \sum_{k=1}^{K_2} \beta_k S_{i,j}^{(g_k)} & i = j \\ - \sum_{k=1}^{K_2} \beta_k S_{i,j}^{(g_k)} & i \neq j \end{cases}$$

Slično se dobije i oblik funkcije Ψ koji utiče na smanjenje broja lokalnih maksimuma u skupu izlaznih vrednosti y_1, \dots, y_t :

$$- \sum_{i=1}^t \sum_{j=1}^t \sum_{k=1}^{K_2} \gamma_k \frac{1}{2} S_{i,j}^{(l_k)} (y_i + y_j)^2 = -\mathbf{y}^T \mathbf{C} \mathbf{y} \quad (2.52)$$

gde je matrica \mathbf{C} definisana kao:

$$C_{i,j} = \begin{cases} \sum_{k=1}^{K_3} \gamma_k \sum_{r=1}^t S_{i,r}^{(l_k)} + \sum_{k=1}^{K_3} \gamma_k S_{i,j}^{(l_k)} & i = j \\ \sum_{k=1}^{K_3} \gamma_k S_{i,j}^{(l_k)} & i \neq j \end{cases}$$

Ako uvedemo matricu $\Sigma^{-1} = 2(A + B + C)$ tada funkciju Ψ možemo predstaviti u obliku:

$$\Psi = -\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y} + \mathbf{y}^T \mathbf{d} - c$$

Formula 2.44 tada ima oblik:

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y} + \mathbf{y}^T \mathbf{d}} e^{-c}}{e^{-c} \int_{-\infty}^{+\infty} e^{-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y} + \mathbf{y}^T \mathbf{d}} d\mathbf{y}} \quad (2.53)$$

gde je faktor normalizacije jednak:

$$\int_{-\infty}^{+\infty} e^{-\frac{1}{2}\mathbf{y}^T \Sigma^{-1} \mathbf{y} + \mathbf{y}^T \mathbf{d}} d\mathbf{y} = \frac{(2\pi)^{\frac{n}{2}}}{|\Sigma^{-1}|^{\frac{1}{2}}} e^{\frac{1}{2}\mathbf{d}^T \Sigma \mathbf{d}} = \frac{(2\pi)^{\frac{n}{2}}}{|\Sigma^{-1}|^{\frac{1}{2}}} e^{\frac{1}{2}\boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu}} \quad [27]$$

Pri tom je $\boldsymbol{\mu} = \Sigma \mathbf{d}$.

Korišćenje činjenice $|\Sigma^{-1}| = \frac{1}{|\Sigma|}$ i sređivanjem 2.53 dobijamo

$$p(\mathbf{y}|\mathbf{X}) = \frac{e^{-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{y}-\boldsymbol{\mu})}}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \quad (2.54)$$

što je i bilo potrebno dokazati.

■

Prethodna teorema nam omogućava lako izvođenje parcijalnih izvoda neophodnih za treniranje modela i direktno računanje rešenja problema 2.45:

$$\boldsymbol{\mu} = \Sigma \mathbf{d} = \arg \max_{\mathbf{y}} p(\mathbf{y}|\mathbf{X}) \quad (2.55)$$

Iako postoje bolji izbori za funkcije l , jer funkcije l u postojećem obliku kažnjavaju i kada je jedna vrednost y_i velika a druga y_j mala, menjanjem oblika funkcije l ne bi mogli da izvedemo gausovu raspodelu u prethodnoj teoremi. To bi uvelo komplikacije pri izvođenju oblika parcijalnih izvoda i dodatani nivo optimizacije.

Kako bi da imenilac formule 2.44 konvergirao, neophodno je ograničiti parametre modela: $\alpha_k > 0$, $\beta_k > 0$, $\gamma_k > 0$ dok θ_k nema ograničenja. Kako je poželjno da optimizacija nema ograničenja nad parametrima nad kojima optimizujemo, računamo parcijalne izvode $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \alpha_k}$, $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \beta_k}$ i $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \gamma_k}$ koristeći $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \alpha_k}$, $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \beta_k}$ i $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \gamma_k}$.

$$\begin{aligned} \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \alpha_k} &= \alpha_k \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \alpha_k} - \lambda_\alpha \alpha_k^2 \\ \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \beta_k} &= \beta_k \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \beta_k} - \lambda_\beta \beta_k^2 \\ \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \ln \gamma_k} &= \gamma_k \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \gamma_k} - \lambda_\gamma \gamma_k^2 \end{aligned} \quad (2.56)$$

Prethodne formule su regularizovane koristeći parametre λ_α , λ_β i λ_γ .

Ovde ćemo prikazati formule, prethodno navedenih parcijalnih izvoda koji se javljaju sa desne strane u jednakostima 2.56, bez izvođenja. Za izvođenje formula

pogledati rad Baltrušaitis-Banda-Robinson [4] gde je pokazano kako se formule izvode.²⁰

Neka su matrice $B^{(k)}$ i $C^{(k)}$ definisane kao:

$$(B^{(k)})_{ij} = \begin{cases} \sum_{r=1}^t S_{i,r}^{(g_k)} - S_{i,j}^{(g_k)} & i = j \\ -S_{i,j}^{(g_k)} & i \neq j \end{cases} \quad \text{i} \quad (C^{(k)})_{ij} = \begin{cases} \sum_{r=1}^t S_{i,r}^{(l_k)} + S_{i,j}^{(l_k)} & i = j \\ S_{i,j}^{(l_k)} & i \neq j \end{cases} \quad (2.57)$$

i neka je $h(\Theta^T \mathbf{X})_{k,*}$ k -ta vrsta matrice $h(\Theta^T \mathbf{X})$, tada je:

$$\begin{aligned} \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \alpha_k} &= -\mathbf{y}^T \mathbf{y} + 2\mathbf{y}^T h(\Theta^T \mathbf{X})_{k,*}^T - 2h(\Theta^T \mathbf{X})_{k,*} \boldsymbol{\mu} + \boldsymbol{\mu}^T \boldsymbol{\mu} + \text{tr}(\Sigma) \\ \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \beta_k} &= -\mathbf{y}^T B^{(k)} \mathbf{y} + \boldsymbol{\mu} B^{(k)} \boldsymbol{\mu} + \text{Vec}(\Sigma)^T \text{Vec}(B^{(k)}) \\ \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \gamma_k} &= -\mathbf{y}^T C^{(k)} \mathbf{y} + \boldsymbol{\mu} C^{(k)} \boldsymbol{\mu} + \text{Vec}(\Sigma)^T \text{Vec}(C^{(k)}) \end{aligned} \quad (2.58)$$

Pri tom je $\text{Vec}(\Sigma)$ operacija vektorizacije matrice Σ koja matricu Σ transformiše u vektor kolone.

Ostalo je jos da se izračuna parcijalni izvod $\frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \theta_k}$. Kako je $\frac{\partial \Sigma^{-1}}{\partial \theta_k} = \frac{\partial 2(A+B+C)}{\partial \theta_k} = 0$ dobijamo:

$$\begin{aligned} \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \theta_k^{(i)}} &= \mathbf{y}^T \frac{\partial \mathbf{d}}{\partial \theta_k^{(i)}} - \frac{1}{2} \frac{\partial \mathbf{d}^T \mathbf{d}}{\partial \theta_k^{(i)}} \\ \frac{\partial \mathbf{d}^T \Sigma \mathbf{d}}{\partial \theta_k^{(i)}} &= \frac{\partial \mathbf{d}^T}{\partial \theta_k^{(i)}} \Sigma \mathbf{d} + \mathbf{d}^T \left(\frac{\partial \Sigma}{\partial \theta_k^{(i)}} \mathbf{d} + \Sigma \frac{\partial \mathbf{d}}{\partial \theta_k^{(i)}} \right) = 2 \frac{\partial \mathbf{d}^T}{\partial \theta_k^{(i)}} \boldsymbol{\mu} \\ \frac{\partial \mathbf{d}^T}{\partial \theta_k^{(i)}} &= \frac{\partial 2\boldsymbol{\alpha}^T h(\Theta^T \mathbf{X})}{\partial \theta_k^{(i)}} \\ &= 2\alpha_k [h(\boldsymbol{\theta}_k, \mathbf{x}_1)(1 - h(\boldsymbol{\theta}_k, \mathbf{x}_1))x_1^{(i)}, \dots, h(\boldsymbol{\theta}_k, \mathbf{x}_t)(1 - h(\boldsymbol{\theta}_k, \mathbf{x}_t))x_t^{(i)}] \end{aligned} \quad (2.59)$$

Iz prethodnog sledi:

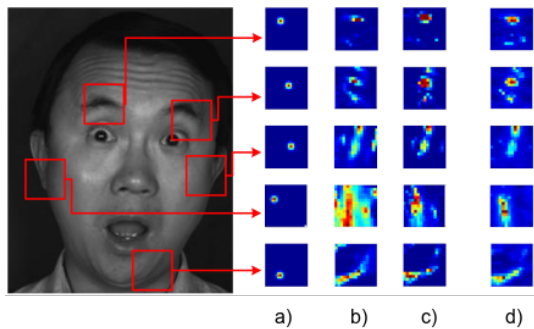
$$\begin{aligned} \frac{\partial \ln p(\mathbf{y}|\mathbf{X})}{\partial \theta_k} &= \begin{bmatrix} \left(\frac{\partial \mathbf{d}^T}{\partial \theta_k^{(1)}}\right)_{1,1} & \cdots & \left(\frac{\partial \mathbf{d}^T}{\partial \theta_k^{(1)}}\right)_{t,1} \\ \vdots & \ddots & \vdots \\ \left(\frac{\partial \mathbf{d}^T}{\partial \theta_k^{(s)}}\right)_{1,1} & \cdots & \left(\frac{\partial \mathbf{d}^T}{\partial \theta_k^{(s)}}\right)_{t,1} \end{bmatrix} (\mathbf{y} - \boldsymbol{\mu}) \\ &= 2\alpha_k \mathbf{X} (h(\Theta^T \mathbf{X})_{k,*}^T \odot (J_{n,1} - h(\Theta^T \mathbf{X})_{k,*}^T) \odot (\mathbf{y} - \boldsymbol{\mu})) \end{aligned} \quad (2.60)$$

gde je \odot oznaka za Adamarov proizvod, a $J_{n,1}$ vektor kolona jedinica dimenzije n .

Koristeći formule 2.56, 2.58 i 2.60 kao i BFGS metod optimizacije [23] dobijamo rešenje problema 2.50.

Vizualizacija rešenja eksperta polja, data formulom 2.45, za argument $X^{(l)}$ je data slikom 11 (crvenija boja odgovara većoj verovatnoći da je ključna tačka poravnata na toj lokaciji). U prethodnom $X^{(l)}$ je matrica koja u i -toj koloni sadrži element $x_i \in \mathcal{R}^s$,

²⁰Uz napomenu da je u [4] matrica $\Sigma^{-1} = 2(A+B)$ a za funkciju Ψ definisanu sa 2.46 treba koristiti $\Sigma^{-1} = 2(A+B+C)$.



Slika 2.10: Primer rada određenih eksperata polja.

- a) Originalne pozicije ključnih tačaka.
 b) Primer rada eksperta polja baziranog na logističkom regresoru. Primer rada takvog regresora dat je na [slici 2.5A](#).
 c) Eksperti polja gde funkcija Ψ ne sadrži sabirke koji sadrže funkcije g ili l (takozvane ivične atribute).
 d) Kompletni ekspert polja dat formulom [2.46](#).

gde je x_i piksel koji pripada kvadratnom regionu (sa centrom u trenutnoj proceni l -te ključne tačke) u kojem se procenjuje vrednost eksperta polja.

2.3 Pronalaženje ekstremuma signala

Signal formiran od određenih podataka često sadrži šum koji potiče od tih podataka, stoga neće svi lokalni maksimumi biti tačke od interesa. Iz tog razloga često se koriste posebni algoritmi koji detektuju ekstremume signala, izuzimajući veliki broj lokalnih maksimuma nastalih šumom. Iako je veoma jednostavno vizualno locirati ekstremume signala koji zavise od jedne promenljive, ipak je korisno uopštiti algoritam koji ima za cilj pronalaženje takvih ekstremuma kako bi izbegli subjektivnost i ubrzali proces njihovog pronalaženja. Takav algoritam pokazao se kao neophodan ako želimo da razvijemo sistem za detekciju mikroekspresija. Kako bi olakšali formulaciju algoritma koji pronalazi ekstremume od interesa uvešćemo nekoliko osnovnih pojmova.

Neka je $T = \{x_1, \dots, x_n\}$ skup uniformno raspoređenih podataka, zabeleženih u jednako udaljenim vremenskim trenucima $1, \dots, N$. Bilo koji skup $W \subset T$ takav da je $W = \{x_{j-m}, x_{j-m+1}, \dots, x_j, \dots, x_{j+m}\}$ se zove prozorom tačke x_j radijusa m . Na osnovu definicije prozora očigledno je da W sadrži isključivo vremenski uzastopne podatke.

Neka je dato $0 < d < \frac{n}{2}$. Tačka $x^{(l)}$ je lokalni ekstremum od interesa ako važe sledeći uslovi:

- $x^{(l)}$ je lokalni maksimum unutar prozora tačke $x^{(l)}$ radijusa d . Tačka $x^{(l)}$ ne mora biti globalni maksimum skupa T .
- Vrednost tačke $x^{(l)}$ je velika u odnosu na ostale vrednosti u prozoru tačke $x^{(l)}$ radijusa d . Tačka $x^{(l)}$ ne mora biti velika u odnosu na ostale tačke skupa T .
- $x^{(l)}$ mora biti i izolovana, dakle ne sme biti više tačaka slične vrednosti u prozoru tačke $x^{(l)}$ radijusa d .

Lokalni ekstremum od interesa biće globalni ekstremum od interesa ako je njegova vrednost dovoljno velika u odnosu na ostale tačke skupa T .

Lokalne ekstremume nećemo tražiti direktno već ćemo, koristeći funkciju S , svakoj tački x_i dodeliti nenegativan realan broj koji opisuje koliko je vrednost x_i izražena u odnosu na vrednosti prozora tačke x_i radijusa d . Kako je teško uvesti funkciju S koja je jednako dobra u svim domenima primene, ima smisla definisati više oblika funkcije S koje će na različite načine računati koliko je tačka x_i dobar izbor za lokalni ekstremum od interesa.

Neka je $k > 0$. Definišemo skupove $N^+(k, i, T) = \{x_{i+1}, x_{i+2}, \dots, x_{i+k}\}$, $N^-(k, i, T) = \{x_{i-k}, x_{i-k+1}, \dots, x_{i-1}\}$. Neka je funkcija $D_{x_i}(F) = \{x_i - y | y \in F\}$ gde je F neki skup. Tada za funkciju S možemo uzeti neku od sledećih funkcija:

$$\begin{aligned} S_1(k, i, T) &= \frac{\max D_{x_i}(N^-(k, i, T)) + \max D_{x_i}(N^+(k, i, T))}{2} \\ S_2(k, i, T) &= \frac{\text{avg } D_{x_i}(N^-(k, i, T)) + \text{avg } D_{x_i}(N^+(k, i, T))}{2} \\ S_3(k, i, T) &= \frac{(x_i - \text{avg } N^-(k, i, T)) + (x_i - \text{avg } N^+(k, i, T))}{2} \\ S_4(k, w, i, T) &= H_w(N^-(k, i, T) \cup x_i \cup N^+(k, i, T)) \\ &\quad - H_w(N^-(k, i, T) \cup N^+(k, i, T)) \end{aligned} \quad (2.61)$$

gde je $\text{avg } F$ prosek elemenata skupa F , a $H_w(F)$ funkcija koja računa entropiju elemenata skupa F .

Za $F = \{a_1, \dots, a_M\}$ funkcija H_w je data u obliku:

$$H_w(F) = \sum_{i=1}^M -p_w(a_i) \ln p_w(a_i) \quad (2.62)$$

Procena vrednosti gustine verovatnoće u tački a_i data je sa $p_w(a_i)$:

$$p_w(a_i) = \frac{1}{M|a_i - a_{i+w}|} \sum_{j=1}^M K\left(\frac{a_i - a_j}{|a_i - a_{i+w}|}\right) \quad (2.63)$$

gde je funkcija K bilo koja funkcija koja ispunjava uslove:

- K je integrabilna i važi $\int_{-\infty}^{+\infty} K(u) du = 1$
- K je nenegativna, $K(u) \geq 0$
- K je simetrična, $K(-u) = K(u)$

Posle računanja vrednosti svih elemenata skupa $S(T)$, računamo prosek m i standardnu devijaciju σ svih pozitivnih vrednosti elemenata skupa $S(T)$.²¹ Globalne ekstremume od interesa tada dobijamo odbacivanjem svih lokalnih ekstremuma od interesa čija je vrednost manja od h standardnih devijacija od proseka.

Primer rada **algoritma 2** za različite odabire funkcije S se može videti na **Slici 2.11**.

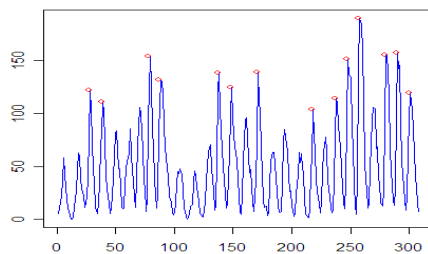
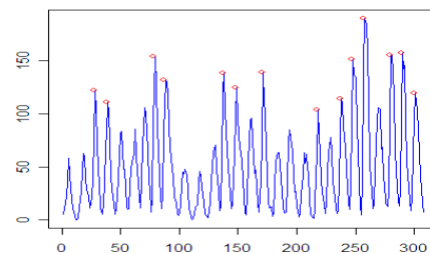
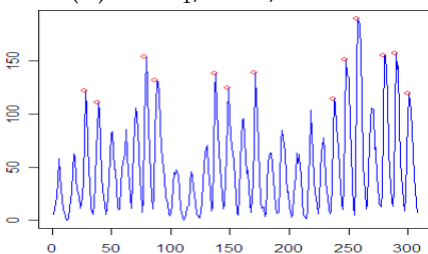
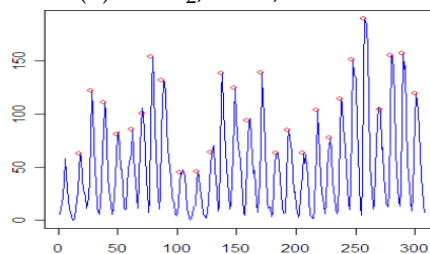
²¹Gde je $S(T) = \{S(k, 1, T), S(k, 2, T), \dots, S(k, n, T)\}$

Algoritam 2: Algoritam za pronalaženje značajnih ekstremuma funkcije

```

Data:  $k, h, T$  (i  $w$  ako je  $S = S_4$ )
Result: Niz elemenata  $O$  koji sadrži ekstremume od interesa
 $O = \emptyset$ ;
// Računamo vrednost funkcije  $S$  za svaku tačku skupa  $T$ 
for ( $i = 0; i < |T|; i++$ ) do
  |  $a[i] = S(k, i, T)$ ;
end
Računamo prosek  $m$  i standardnu devijaciju  $s$  svih pozitivnih elemenata niza
 $a$ ;
// Na osnovu lokalnih ekstremuma od značaja dobijamo globalne
// tako što odbacujemo male vrednosti.
for ( $i = 0; i < |T|; i++$ ) do
  | if ( $a[i] > 0 \wedge (a[i] - m) > (h * s)$ ) then
    | |  $O = O \cup \{i\}$ ;
  | end
end
// Ukoliko postoji prozor radijusa  $k$  koji sadrži dve vrednosti
// skupa  $O$  eliminišemo manju vrednosti iz skupa  $O$ 
for ( $i = 1; i < |O|; i++$ ) do
  | if ( $(O[i] - O[i - 1]) \leq k$ ) then
    | | Uklanjamo indeks elementa  $\min\{T[O[i]], T[O[i - 1]]\}$  iz  $O$ ;
    | |  $i--$ ;
  | end
end
return  $O$ ;

```

(A) $S = S_1, k = 5, h = 1.5$ (B) $S = S_2, k = 5, h = 1.5$ (C) $S = S_3, k = 5, h = 1.5$ (D) $S = S_4, w = 5, k = 5, h = 1.5$

Slika 2.11: Rezultat primene algoritma 2 za različite odabire funkcije S . Skup podataka T predstavlja broj sunčevih pega tokom jedne godine počev od 1700. godine pa do 2008. godine.

2.4 Transformacija i izdvajanje lica

Kako se lica tokom snimanja video kamerom mogu pomerati²², veoma je teško raditi analizu mikroekspresija nad originalnim video zapisom. Neophodno je da neutrališemo pomeraje lica kako bi radili bilo kakve smislene analize u svrhu pronalazjenja mikroekspresija.

Kako bi neutralisali pomeraje lica potrebno je da pronađemo lokacije ključnih tačaka lica te slike (str 8). Zatim izvajamo tri ključne tačke koje odgovaraju lokacijama unutrašnjih uglova očiju kao i koščici koja se nalazi u podnožju nosa (Pogledati sliku 13).²³ Prethodno navedene tri ključne tačke su veoma pogodan izbor iz dva razloga:

- Unutrašnji uglovi očiju i podnožje nosa se ne mogu značajno pomeriti menjanjem izraza lica
- Algoritam za detekciju i praćenje ključnih tačaka nalazi prethodno navedene ključne tačke sa veoma malom prostornom greškom

Neka je:

- A = lokacija unutrašnjeg ugla desnog oka.
- B = lokacija unutrašnjeg ugla levog oka.
- C = središte duži AB

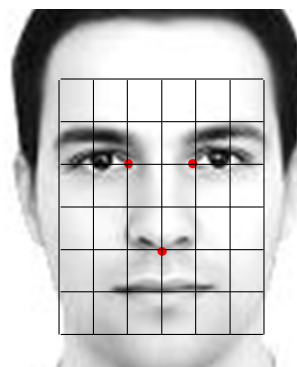
tada je $\theta = \angle(AB, x \text{ osa})$. Rotacijom svih tačaka slike za ugao θ sa centrom u C ako je A iznad B , odnosno $-\theta$ ako je B iznad A dobijamo sliku lica na kome su unutrašnji uglovi očiju paralelni sa x -osom (slika 2.12).

Novodobijenu sliku lica zatim delimo na 36 pravougaonih oblasti jednake veličine kao što je prikazano na slici 2.12. Podela lica na blokove se radi kako bi izdvojili informacije o teksturi svakog bloka ponaosob [1].

Informacije o teksturi bloka će se pokazati bitnim za detekciju mikroekspresija jer će promena izraza lica često rezultovati promenom tekstone lica u nekom od blokova. Eksperimentalno je pokazano da je podela lica na blokove, prikazana na slici 2.12, najbolja za potrebe detekcije mikroekspresija[21].

Pri određivanju veličine bloka posmatra se rastojanje između unutrašnjih uglova očiju kao i rastojanje od podnožja nosa do linije koje formiraju unutrašnji uglovi očiju. Ove veličine mogu varirati od osobe do osobe [16] pa je neophodno izmeriti data odstojanja za svaku osobu ponaosob kako bi obezbedili da površina koju pokrivaju blokovi obuhvate sve bitne oblasti lica.²⁴

Iako smo izvršili neophodnu rotaciju (i eventualno skaliranje) nad slikama originalnog video zapisa, potrebno je još ukloniti pozadinu sa slike kako bi analiza lica, za potrebe detekcije mikroekspresija, bila još stabilnija. Kako bi izdvojili lice sa slike radi se triangulacija nad skupom ključnih tačaka kao što je prikazano na Slici 2.13. Svi pikseli slike koji ne pripadaju unutrašnjoj oblasti određenoj triangulacijom površi definisane ključnim tačkama lica se boje jednom bojom [19].

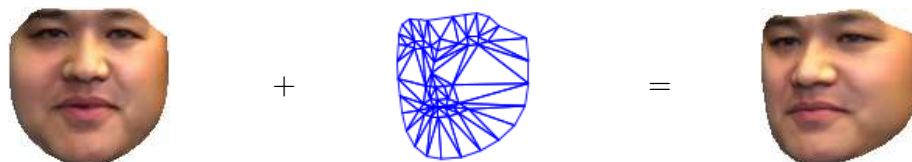


Slika 2.12: Tri bitne ključne tačke i podela lica na blokove definisanim tim ključnim tačkama

²²Konkretno mogu se rotirati ili približavati/odaljavati od kamere

²³Latinski naziv za datu koščicu u podnožju nosa je 'Spina nasalis anterior maxillae'

²⁴Bitne za potrebe detekcije mikroekspresija



Slika 2.13: Triangulacija oblasti lica na osnovu lokacija ključnih tačaka i izdvajanje lica iz slike nezavisno od položaja glave.

2.5 Lociranje mikroekspresija

Kako bi jednostavnije objasnili proces lociranja mikroekspresija uvešćemo nekoliko korisnih pojmova. Neka je CF trenutna slika video zapisa koja se obrađuje. TF je slika koja se nalazi k slika iza CF a HF slika koja se nalazi k slika ispred CF. Ukupan broj slika od TF do HF (uključujući TF i HF) je N .²⁵

N dobijamo iz činjenice da mikroekspresija ne može biti duža od x sekundi. Ako video sadrži s slika u jednoj sekundi prikazivanja tada je N prvi neparan broj veći ili jednak od xs . Uglavnom važi da je $x \leq 0.5$.

Slike CF, TF i HF su u procesu pretprocesiranja podeljene na blokove kao što je objašnjeno u delu "Transformacije i izdvajanje lica". Neka su redom CF_i^{LBP} , HF_i^{LBP} , TF_i^{LBP} izračunate vrednosti LBP atributa slika CF, HF, TF za i -ti blok. Način na koji se blokovi indeksiraju su prikazani na slici 2.14.

Računanjem normalizovanih histograma LBP atributa datih sa CF_i^{LBP} , HF_i^{LBP} , TF_i^{LBP} dobijamo redom CF_i^{HIS} , HF_i^{HIS} i TF_i^{HIS} . Označimo sa AFF_i^{HIS} prosek histograma HF_i^{HIS} i TF_i^{HIS} . Osnovna ideja je da velike razlike između histograma CF_i^{HIS} i AFF_i^{HIS} predstavljaju indikator da se desila promena u i -tom bloku lica. Takođe ta promena je brza jer se desila između HF i TF, a mikrinterval N se bira tako da je vremenski kratak.²⁶

Ovo možemo intuitivno lako potvrditi na primerima mikroekspresije i makroekspresije. Primer makroekspresije osmeha bila bi osoba koja kreće da prikazuje osmeh između TF i CF i održava osmeh na licu i između CF i HF. Vizualizacija ovog primera bila bi plava kriva na slici 2.15.

Primer mikroekspresije osmeha bi bio ako osoba počinje da prikazuje osmeh između TF i CF ali je već između CF i HF osmeh potisnut. Ovakav scenario bi odgovarao crvenoj krivoj na slici 2.15. Očigledno je da će, u proseku, razlika histograma CF_i^{HIS} i AFF_i^{HIS} biti veća na primeru mikroekspresije nego na primeru makroekspresije.

Pri tom je na slici 2.15 problem prikazan u jednoj dimenziji (prikazane su vrednosti samo jedne klase histograma) što nam, iako je problem značajno pojednostavljen, i dalje omogućava da razumemo proces iza analize mikroekspresija.

Razliku između normalizovanih histograma P i Q sa istim brojem klasa (podeoka) računamo pomoću χ^2 razlike:

$$\chi^2(P, Q) = \sum_i \frac{(P_i - Q_i)^2}{P_i + Q_i}$$

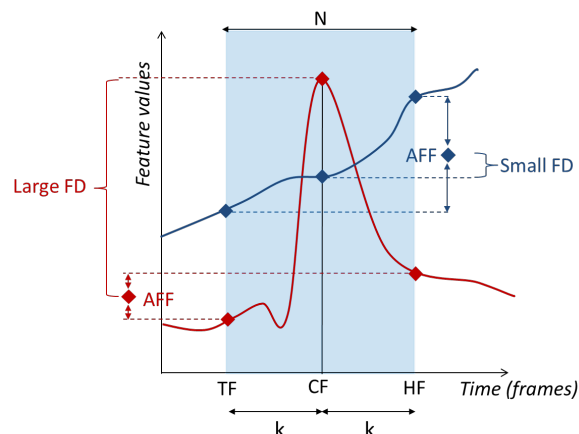
Neka je CF j -ta slika po redu u video zapisu koji se analizira. Iz prethodnog primera je jasno da je korisno izračunati $d_{j,i} = \chi^2(CF_i^{HIS}, AFF_i^{HIS})$ za svaki blok $i \in \{0, 1, \dots, 35\}$.

²⁵Očigledno je da je N neparan broj i da važi $k = \frac{N-1}{2}$.

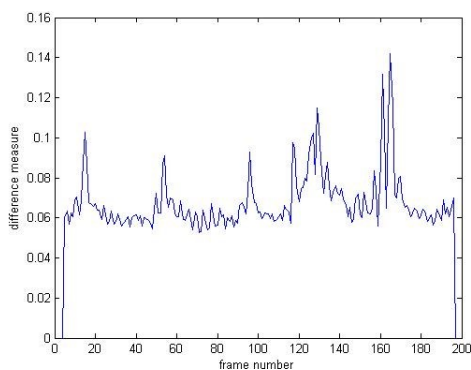
²⁶Ovde zapravo biramo malu vrednost za x - dužinu trajanja mikroekspresije.



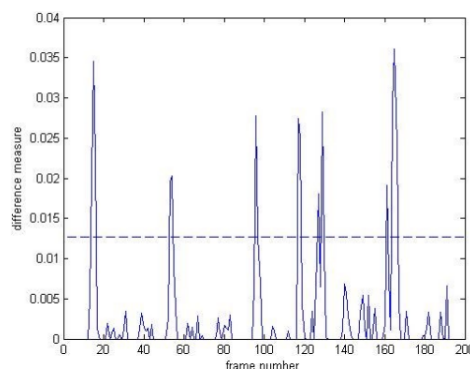
Slika 2.14: Indeksiranje blokova lica



Slika 2.15: Vizualizacije analize slika u potrazi za mikroekspresijama.



Slika 2.16: Primer vrednosti F atributa dobijenih pri analizi video zapisa.



Slika 2.17: Primer vrednosti C atributa dobijenih pri analizi video zapisa, uz označenu graničnu vrednost T .

Označimo sa $d_j^* = \{d_{j,0}^*, d_{j,1}^*, \dots, d_{j,35}^*\}$ niz $d_j = \{d_{j,0}, d_{j,1}, \dots, d_{j,35}\}$ sa vrednostima sortiranim u opadajućem redosledu. Tada se F -atribut j -te slike video zapisa označava sa F_j i definisan je sa:

$$F_j = \frac{1}{M} \sum_{i=0}^{M-1} d_{j,i}^* \quad (2.64)$$

Za F_j možemo da smatramo da meri odudaranje j -te slike od "proseka" okolnih slika.²⁷

Broj M određuje koliko najvećih razlika blokova se uzima u obzir. Eksperimentalno je pokazano da se bolji rezultati postižu ako se razmatra trećina najvećih razlika blokova nego u slučaju kada bi posmatrali maksimum razlika blokova ili prosek razlika svih blokova [21]. Stoga će, za podelu lica na 36 blokova, uglavnom važiti da je $M = 12$. Primer grafika funkcije F -atributa dobijen pri analizi video zapisa dat je na slici 2.16.

²⁷Zapravo merimo odudaranje histograma LBP atributa slike CF (u ovom slučaju j -ta slika video zapisa) od proseka histograma LBP atributa okolnih slika TF i HF, ali je smisao isti - detekcija kratkih pokreta.

Problem koji se javlja kod računanja F atributa je taj što se ne mogu izračunati za prvih i poslednjih k slika video zapisa. Ako bi CF bila neka od prvih k slika video zapisa tada bi TF bila slika koja se nalazi pre prve slike, slično bi važno i za HF i poslednjih k slika. Jedan način da se ovo razreši jeste da kad god se TF nalazi pre prve slike za TF odaberemo prvu sliku video zapisa. Analogno se za HF bira poslednja slika u slučaju kada se HF nalazi nakon poslednje slike. Ovakav odabir slika za TF i HF nije idealan pa je najbolje, kad god je to moguće, ne računati F attribute za prvih i poslednjih k slika video zapisa.

Kako bi, među vrednostima F atributa, razlikovali relevantne ekstremume od onih uzrokovanih šumom od svakog F atributa slike CF oduzećemo prosek F atributa slika HF i TF. Na taj način formiramo C atribut koji se za j -tu sliku video zapisa definiše kao:

$$C_j = \begin{cases} F_j - \frac{1}{2}(F_{j-k} + F_{j+k}) & F_j - \frac{1}{2}(F_{j-k} + F_{j+k}) \geq 0 \\ 0 & \text{inače} \end{cases}$$

Slučaj kada je $F_j - \frac{1}{2}(F_{j-k} + F_{j+k}) < 0$ znači da je F atribut slike CF manji nego prosek F atributa njene okoline, što znači da pokreti u slici CF nisu brzi uzimajući u obzir njene okolne slike HF i TF. Primer grafika funkcije C atributa, dobijenih od F atributa prikazanih na slici 2.16, je prikazan na slici 2.17.

I ovde postoji sličan problem kao i kod F atributa, a to je da C atributi ne postoje u slučaju prvih i poslednjih $2k$ slika video zapisa.²⁸ Ovo nije nužno veliki problem jer možemo smatrati da subjekat, pri snimanju, neće prikazati mikroekspresiju u prvih i poslednjih $2k$ slika.²⁹

Nakon izračunavanja C atributa za sve slike video zapisa za koje je to izvodljivo, pronalazimo ekstremume u skupu vrednosti C atributa, kao što je objašnjeno u delu 2.3. Sve ekstremume manje od granične vrednosti T odbacujemo, gde je T definisano kao:

$$T = \bar{C} + \tau(C_{max} - \bar{C}) \quad \tau \in [0, 1] \quad (2.65)$$

Pri tom je \bar{C} prosek C atributa, a C_{max} maksimalna vrednost C atributa. Parametar τ se zadaje, gde za $\tau = 0$ uzimamo sve ekstremume veće ili jednake od proseka C atributa, a za $\tau = 1$ uzimamo u obzir samo najveću vrednost C atributa.

Veća vrednost parametra τ rezultuje većom verovatnoćom da propustimo mikroekspresiju pri analizi, ali i manjom verovatnoćom da zabeleženi ekstremum C atributa ne odgovara mikroekspresiji. Slično, manje τ će rezultovati većom šansom da ekstremum C atributa ne odgovara mikroekspresiji, ali i manjom verovatnoćom da ćemo propustiti neku mikroekspresiju pri analizi.

Označimo skup preostalih vrednosti ekstremuma³⁰ C atributa sa $P = \{C_{j_0}, C_{j_1}, \dots, C_{j_m}\}$, gde je j_i redni broj slike u video zapisu čija je vrednost C atributa C_{j_i} . Postavićemo uslov da dve mikroekspresije moraju biti udaljene barem $\frac{k}{2} + 1$ sliku. Ako postoje C_{j_i}, C_{j_k} tako da važi $|j_i - j_k| \leq \frac{k}{2}$ iz P izbacujemo manju od vrednosti C_{j_i} i C_{j_k} . Prethodni korak se ponavlja sve dok u P ne postoji par (C_{j_i}, C_{j_k}) za koji važi $|j_i - j_k| \leq \frac{k}{2}$. Za slike, čija vrednost C atributa pripada novodobijenom skupu P , smatramo da sadrže izraz mikroekspresije u njihovom najvećem intenzitetu.

²⁸ F atributi ne postoje za prvih i poslednjih k slika video zapisa, a C atributi ne postoje u slučaju prvih i poslednjih k F atributa.

²⁹Dovoljno je da ne postavljamo pitanja subjektu u prvih $2k$ slika i da nakon poslednjeg odgovora sačekamo još $2k$ slika pre nego što ugasimo kameru.

³⁰Svi ekstremumi C atributa čija je vrednost veća ili jednaka od T

Glava 3

Program za detekciju mikroekspresija

3.1 Nacrt programa

Pri konstrukciji programa za detekciju mikroekspresija vodićemo se definicijom mikroekspresije. Naime, kao što je već rečeno, mikroekspresija je kratak, nekontrolisan izraz lica koji se javlja kod ljudi i koji sa sobom nosi informaciju o određenoj emociji.

Program za detekciju mikroekspresija ne mora (i neće) registrovati makroekspresije¹, ali zato mora biti u stanju da detektuje kratke i brze pokrete lica. To se postiže upoređivanjem uzastopnih slika video zapisa unutar određenog vremenskog intervala.

Za metodu koja će biti predstavljena nije neophodno nikakvo dodatno učenje parametara niti prethodno označavanje ulaznih video zapisa. Ukratko, metoda se sastoji iz 5 delova:

- Nalazimo i pratimo ključne tačke lica prikazanih u video zapisu.
- Na osnovu određenih mera na licu, oblast lica se deli na uzajamno jednake blokove čija veličina zavisi od oblika glave (lica).
- Računaju se LBP atributi za svaki blok.
- Računaju se razlike LBP atributa odgovarajućih blokova uzastopnih slika video zapisa.
- Uzimajući prosek određenog broja najvećih razlika atributa blokova i koristeći algoritam za pronalaženje ekstremuma signala dobijamo informaciju o tome koje slike video zapisa sadrže mikroekspresije u njihovom najvećem intenzitetu.

¹Makroekspresije su promena izraza lica u trajanju od 0.5s-4s. Makroekspresije vidamo svakodnevno u interakciji sa ljudima.

3.1.1 OpenFace biblioteka

OpenFace biblioteka² omogućava detekciju ključnih tačaka opisanih u delu 2.2. Biblioteka ujedno nudi i dodatne mogućnosti kao što su detekcija akcija određenih mišića kao i praćenje pogleda subjekta [3, 2, 26]. Bitne klase OpenFace biblioteke koje ćemo koristiti su:

- `LandmarkDetector::FaceModelParameters`
Detektor ključnih tačaka će koristiti kao argument instancu ove klase. Pri instanciranju možemo podesiti niz različitih parametara. Podešavanje parametara se radi tako što prosledimo odgovarajuće opcije preko komandne linije (kao dodatne argumente pri pozivu programa).
- `LandmarkDetector::CLNF`
Glavna klasa za detekciju ključnih tačaka, sadrži model oblika lica, eksperte polja kao i implementacije odgovarajućih optimizacionih metoda.
- `FaceAnalysis::FaceAnalyser`
Klasa zadužena za klasifikaciju izraza lica. Izrazi lica se klasifikuju za svaku sliku video zapisa koristeći FACS kodiranje izraza lica[14]. Pri tome ,pri klasifikaciji, nije dat jedinstven kod izraza lica već verovatnoće, za svaki od izraza, da tekuća slika sadrži odgovarajući izraz lica.

Bitni parametri koje možemo podesiti su:

- `-w_reg` - Faktor težine u metodi najmanjih kvadrata sa težinama.
- `-reg` - Faktor težine regularizacionog dela izraza.
- `-validate_detections` - Ako je ova opcija zadata detekcija lica prolazi kroz proces validacije.
- `-n_iter` - Broj iteracija pri optimizaciji
- `-q` - Ako je ova opcija zadata neće biti izlaznih informacija tokom obrade ključnih tačaka kao i njihove vizualizacije.
- `-wild` - Zadavanjem ove opcije automatski se podešava niz različitih parametara za potrebe detekcije ključnih tačaka lica u uslovima koji nisu idealni.

Funkcija koja omogućava detekciju ključnih tačaka u video zapisu je:

```
bool DetectLandmarksInVideo(const cv::Mat_<uchar> &grayscale_image, CLNF&
    ↪ clnf_model, FaceModelParameters& params)
```

Argumenti funkcije su, redom, trenutna slika video zapisa u nijansama sive, instanca CLNF klase i parametri modela (instancija `FaceModelParameters` klase). Funkcija vraća `true` ako je detekcija ključnih tačaka uspešna, inače vraća `false`.

²Tokom pisanja ovog rada izašla je i verzija 2.0 OpenFace biblioteke. Za potrebe ovog rada korišćena je verzija 1.0

3.2 Inicijalizacija parametara

Pre bilo kakve obrade neophodno je inicijalizovati parametre. Pri tom parametri mogu biti parametri kamere, ulazno izlazni parametri, parametri modela lica ili parametri sistema za detekciju mikroekspresija. U slučaju parametara kamere ili ulazno izlaznih parametara, instanca klase `CLNFIO` podešava i održava parametre. Za podešavanje tih parametara koristi se konstruktor klase `CLNFIO`:

```
CLNFIO::CLNFIO(std::vector<std::string>& arguments)
```

Bitni ulazno-izlazni parametri koje možemo proslediti klasi `CLNFIO` su:

- `-root` - Početna putanja svih datoteka
- `-inroot` - Početna putanja svih ulaznih datoteka
- `-outroot` - Početna putanja svih izlaznih datoteka
- `-f` - Putanja do ulazne video datoteke
- `-of` - Putanja izlazne tekstualne datoteke koja za svaku sliku sadrži podatke o tačnosti detekcije, izrazu lica, položaju glave,...
- `-ov` - Putanja do izlazne video datoteke detektora ključnih tačaka
- `-oc` - Kodek izlazne video datoteke
- `-visualisefeatures` - Zadavanjem ove opcije omogućavamo prikaz ključnih tačaka u izlaznoj video datoteci
- `-novidshow` - Ako je ova opcija zadata neće biti prikazan izlazni video
- `-nosimsave` - Ako je ova opcija zadata poravnate slike (odnosno slike na kojima je moguće meriti sličnosti lica) neće biti sačuvane
- `-nosimshow` - Ako je ova opcija zadata poravnate slike neće biti prikazane
- `-simalign` - Putanja do direktorijuma u kojem će biti sačuvane poravnate slike

Parametri kamere koje možemo proslediti klasi `CLNFIO` su:

- `-fx` - Fokalna dužina po x osi
- `-fy` - Fokalna dužina po y osi
- `-cx` - x koordinata optičkog centra (principijalne tačke kamere)
- `-cy` - y koordinata optičkog centra (principijalne tačke kamere)
- `-device` - ID uređaja kamere

Za podešavanje i čuvanje parametara modela lica zadužena je instanca klase `FaceModelParameters`. Slično kao i kod klase `CLNFIO`, za postavljanje parametara koristi se konstruktor klase. Parametri koji se mogu proslediti toj klasi su:

- `-mloc` - Putanja do sačuvanog modela za detekciju ključnih tačaka
- `-fdloc` - Putanja do sačuvanog modela za detekciju lica
- `-sigma` - Vrednost σ se koristi za kontrolisanje glatkosti aproksimacije funkcija eksperata polja

- `-w_reg` - Faktor težine kod metode najmanjih kvadrata sa težinama
- `-reg` - Regularizacioni faktor
- `-multi_view` - Ako smo prosledili 1 uz ovaj parametar, smatramo da slike lica nisu isključivo u anfasu. Tada se procena ključnih tačaka vrši za nekoliko pretpostavljenih orijentacija glave i vraća se najbolja procena. Ako je slika lica u anfasu prosleđujemo 0 uz parametar.
- `-n_iter` - Broj iteracija pri optimizaciji formule 2.13
- `-gaze` - Ako je ovaj parametar zadat, tokom analize video zapisa, prati se i pravac pogleda osobe sa snimka
- `-q` - Zadavanjem ovog parametra aktivira se tihi režim rada. Tada se rezultati i izveštaj ne prikazuju
- `-wild` - Detekcija se uglavnom radi na video zapisima koji nisu savršeni (bilo po pitanju osvetljenja ili šuma). Za takve potrebe zadaje se parametar `-wild` koji detektor ključnih tačaka prilagođava takvim video zapisima

Podešavanja parametara sistema za detekciju mikroekspresija vrše se pomoću instance klase `RM_AnalyzerParams`. Parametri se postavljaju pomoću metode

```
void RM_AnalyzerParams::getParams(std::vector<std::string>& arguments)
```

i oni mogu biti:

- `-fullfeatures` - Ako je ovaj parametar prosleđen računaju se F atributi za sve slike video zapisa
- `-thau` - Vrednost $\tau \in [0, 1]$ iz jednakosti 2.65
- `-timeframe` - Maksimalno trajanje mikroekspresije.
- `-lbp_original` - Ako je ovaj parametar podešen koriste se originalni LBP atributi. Umesto suseda na kružnici posmatraju se susedi iz kvadratnog regiona 3×3 .
- `-lbp_extended` - Postavljanjem ovog parametra podrazumevamo korišćenje LBP atributa opisanih u glavi 2.
- `-nr1bp` - Postavljanjem ovog parametra podrazumevamo korišćenje NRLBP atributa (Noise-Resistant LBP) [25]. Uz parametar potrebno je proslediti i prag nesigurnosti bita (ceo broj)
- `-noshowfeaturegraph` - Ako je ovaj parametar postavljen neće biti prikazani grafici F i C atributa za dati video zapis
- `-lbp_neigh` - Broj suseda LBP atributa
- `-lbp_rad` - Poluprečnik LBP atributa
- `-normed_histograms` - Ako je ovaj parametar postavljen svi histogrami biće normirani
- `-uniform_histograms` - Ako je ovaj parametar postavljen za kreiranje histograma korišće se uniformne vrednosti LBP atributa

- `-use_sobel` - Ako je ovaj parametar postavljen, u koraku pretprocesiranja, na slike se primenjuje Sobelov filter kako bi se pronašle ivice
- `-gaussian_blur` - Postavljanjem ovog parametra, u koraku pretprocesiranja, vrši se Gausovo zamučenje slika. Uz parametar je potrebno proslediti i ceo broj a kako bi se formirala matrica axa sa vrednostima funkcije Gausovog zvona standardne devijacije $0.3(\frac{a-1}{2} - 1) + 0.8$
- `-balanced_initial_features` - Postavljanjem ovog parametra, F atributi dati formulom 2.64, se računaju kao razlika proseka M najvećih χ^2 razlika histograma i proseka K najmanjih χ^2 razlika histograma. Uz parametar je potrebno proslediti i realnu vrednost $\frac{n}{K}$ gde je n broj blokova na koje je slika lica podeljena.

Nakon postavljanja odgovarajućih parametara inicijalizujemo, redom, instance klase: `LandmarkDetector::CLNF`, `RM_Analyzer`, `FaceAnalyser` i `NormalizedFace`. Klasa `NormalizedFace` zadužena je za rotaciju lica i ključnih tačaka. Izlazni video date klase biće video lica gde su oči paralelne sa x osom u svakoj slici video zapisa. Nakon instanciranja odgovarajućih klasa potrebno je pozvati metodu koja postavlja skup parametara sistema za detekciju mikroekspresija

```
void RM_Analyzer::setRMParams(RM_AnalyzerParams &params)
```

i metodu koja kreira instancu one LBP klase koju smo prethodno odabrali parametrima

```
void RM_Analyzer::createLBPFeatureExtractor()
```

3.3 Pretprocesiranje

Pod pretprocesiranjem se misli na detekciju i praćenje ključnih tačaka, deljenje oblasti lica na jednake blokove odgovarajuće veličine kao i računanje LBP atributa za svaki blok. Pretprocesiranje se radi za svaku sliku video zapisa. Prvi korak pretprocesiranja podrazumeva poziv funkcije

```
void NormalizedFace::startNewSequence()
```

koja priprema instancu klase `NormalizedFace` za obradu novog video zapisa.

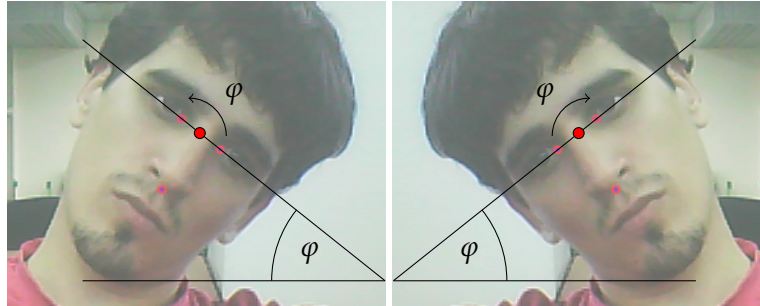
Zatim se poziva metoda `cv::VideoCapture CLNFIO::getNextVideo()` koja omogućava pristup video zapisu koji je sledeći na redu za obradu. Kako bi se izbegao korak pretprocesiranja na onim video zapisima na kojima je već jednom odrađen, poziva se funkcija

```
bool NormalizedFace::existNormalizedVideo(const std::string &path, const
↳ int frames, const double fps)
```

koja proverava da li je video zapis na putanji `path`, koji prikazuje `fps` slika po sekundi i koji sadrži ukupno `frames` slika, prošao korak pretprocesiranja. Ako je to slučaj poziva se metoda

```
void CLNFIO::progress_vid(FaceAnalysis::FaceAnalyser& face_analyser, bool
↳ skipped)
```

koja označava klasi `CLNFIO` da može pripremiti sledeći video zapis za obradu. Argument `skipped` označava da je video zapis preskočen u koraku pretprocesiranja i postavlja se na `true`.

Slika 3.1: Vizualizacija metode `NormalizedFace::addVideoFrame`

U slučaju da video nije prošao korak pretprocesiranja poziva se funkcija

```
void CLNFIO::prepareVidOutput(const cv::Mat& captured_image)
```

koja priprema izlazni video CLNFIO klase. Argument `captured_image` predstavlja sliku iz ulaznog video zapisa na osnovu koje se formira izlazni video zapis.

Nakon pripreme izlaznog video zapisa prelazi se na detekciju i praćenje ključnih tačaka pomoću metode `LandmarkDetector::DetectLandmarksInVideo`. Metoda `LandmarkDetector::DetectLandmarksInVideo` predstavlja implementaciju prethodno opisane metode **regularizovanog prosečnog pomeraja**.

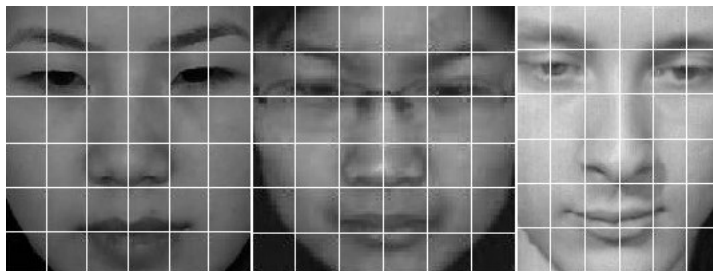
Nakon detekcije ključnih tačaka poziva se metoda `NormalizedFace::addVideoFrame` \rightarrow (`cv::Mat& captured_img`). Metoda `addVideoFrame` formira matricu rotacije za sliku `captured_img` tako da su, primenom te rotacije, oči na slici paralelne sa x -osom. Centar rotacije je lociran u sredini duži određenom unutrašnjim uglovima oka. Matrica rotacije se formira na osnovu ugla definisanim x -osom i pravom koja prolazi kroz unutrašnje uglove oka. Vizualizacija ovog postupka se može videti na **slici 3.1**, a detaljniji opis postupka dostupan je u delu "**Transformacije i izvajanje lica**". Nakon rotiranja slike metoda `addVideoFrame` ažurira instancu CLNF klase koja čuva lokacije ključnih tačaka. Pri ažuriranju CLNF instance vrši se rotacija ključnih tačaka kako bi se promene na slici odrazile na koordinate ključnih tačaka. Ažurirane lokacije ključnih tačaka se zatim koriste za izdvajanje oblasti lica i bojenje pozadine slike jednom bojom. Za izdvajanje lica sa slike i bojenja pozadine koristi se pomoćna klasa `OpenFace` biblioteke (konkretno `FaceAnalysis::PAW` klasa). PAW klasa nam omogućava da izvršimo triangulaciju nad ključnim tačkama lica³, a zatim obojimo jednom bojom sve tačke van površi određene trijangulacijom [19]. Tačke unutar oblasti definisane trijangulacijom nad ključnim tačkama ne menjamo. Ovaj postupak je takođe opisan u delu "**Transformacije i izvajanje lica**".

Nakon što su sve slike video zapisa prosleđene instanci klase `NormalizedFace`, preko metode `addVideoFrame`, poziva se metoda `CLNFIO::progress_vid`, ali ovog puta se argument `skipped` postavlja na `false`, jer video zapis nije preskočen u koraku pretprocesiranja. Poslednji korak pretprocesiranja sastoji se iz poziva metode

```
void NormalizedFace::writeNormalizedVideo(const std::string& path, const
     $\rightarrow$  std::string& output_codec, const bool lines, int base_width)
```

koja koristi slike, prethodno obrađene metodom `addVideoFrame`, i od njih formira video zapis koji sadrži isključivo oblast lica. Video zapis, čije smo slike prosleđivali metodi `addVideoFrame`, se nalazi na putanji `path` gde se takođe čuva i video zapis oblasti lica. Taj video zapis ima isto ime kao i originalni, osim što ima dodat nastavak "`normalized_face`". Oblast lica izlaznog video zapisa je podeljena na 6×6 blokova,

³Pri tom se često lokacije ključnih tačaka koje se vezuju za obrve "podignu" kako bi površ određena triangulacijom ključnih tačaka obuhvatila i deo čela.



Slika 3.2: Deljenje oblasti lica na podudarne blokove.

kao što je to prikazano na slici 3.2. Ako želimo da granične linije blokova budu vidljive na izlaznom video snimku, možemo postaviti `lines` na `true`. Ako želimo da skaliramo oblast lica to možemo učiniti zadavanjem željene širine bloka `base_width` \leftrightarrow . Pri tom, ako je `base_width` negativan broj, lice na izlaznom video snimku će biti iste veličine kao i na ulaznom. Kodek izlazne video datoteke postavlja se pomoću argumenta `output_codec`

3.4 Opis klasa za lociranje mikroekspresija

Za lociranje mikroekspresija zadužena je klasa `RM_Analyzer`. Konstruktor te klase je:

```
RM_Analyzer :: RM_Analyzer ()
```

Prvi korak pri analizi video zapisa, u svrhu traženja mikroekspresija, jeste poziv funkcije

```
void RM_Analyzer :: getNextVid (const std :: string& vid_path)
```

kojim upućujemo instancu klase `RM_Analyzer` na izlazni video klase `NormalizedFace`, generisanim metodom `NormalizedFace :: writeNormalizedVideo`, koji se nalazi na putanji `vid_path`.

Kako ne bi analizirali isti video više puta, za iste parametre klase `RM_Analyzer`, korisno je pozvati funkciju

```
bool RM_Analyzer :: loadParamsIfExistsValidParamsFileForVid (const std ::  
 $\leftrightarrow$  string &path)
```

Prethodna metoda proverava da li je nad video zapisom, na putanji `path`, već rađena analiza pod istim parametrima kao i onim u klasi `RM_Analyzer`. Ako jeste metoda vraća `true`, inače vraća `false`. Ako je metoda vratila vrednost `true` tada je moguće preskočiti analizu video zapisa.

Ako je analiza video zapisa neophodna, potrebno je pozvati metodu `analyzeVid`, koja računa vrednosti F i C atributa. Ako je, preko komandne linije, prosleđen parametar `"-fullfeatures"` tada se računaju F atributi za sve slike video zapisa.⁴

Nakon završene analize video zapisa moguće je pozvati funkciju

```
void RM_Analyzer :: writeParametersToFileForVideo (const std :: string&  
 $\leftrightarrow$  filepath) const
```

koja zapisuje izračunate vrednosti F i C atributa, kao i konfiguraciju detektora pod kojim su te vrednosti izračunate, u tekstualnu datoteku. Putanja `filepath` predstavlja putanju video zapisa nad kojim je rađena analiza.

⁴Za prvih k slika se za TF uzima prva slika, dok se za poslednjih k slika za HF uzima poslednja slika video zapisa

Vizualizacija grafika funkcija F i C atributa, kao i pronalaženje njihovih ekstremuma radi se pomoću funkcije:

```
void drawFeatureDiffs(const bool mark_peaks=true, const bool save=false,
    ↪ const unsigned img_height=400, const unsigned img_width=800, const
    ↪ int x_axis_sections=10, const int y_axis_sections=10) const
```

Argumenti funkcije su:

- `mark_peaks` - Ako je `mark_peaks=true` na slici označavamo ekstremume koji su veći od granične vrednosti T
- `save` - Ako je `save=true` čuva se slika grafika funkcija F i C atributa
- `img_height`, `img_width` - Argumenti koji definišu veličinu slike na kojoj se prikazuje grafik funkcije
- `x_axis_sections`, `y_axis_sections` - Broj podeoka na x , odnosno y osi.

Na kraju analize i detekcije mikroekspresija neophodno je pozvati funkcije `RM_Analyzer` ↪ `::resetAnalyzer`, `FaceAnalyser::Reset` i `CLNF::Reset` koje pripremaju detektor za analizu novog video snimka.

Glava 4

Rezultati programa

4.1 Opis baza podataka

Iako postoji veliki broj baza podataka makroekspresija, one nisu upotrebljive kada je cilj testiranje programa koji detektuje mikroekspresije. Makroekspresije, osim što se razlikuju u dužini trajanja, nose sa sobom i veći intenzitet promene izraza lica u odnosu na spontane mikroekspresije. Mikroekspresije koje nisu spontane (već odglumljene) se takođe razlikuju, kako po trajanju, tako i po intenzitetu od pravih mikroekspresija. Stoga, video datoteke koje čine test skup ne mogu biti ubrzani video zapisi makroekspresija,¹ kao ni video zapisi odglumljenih mikroekspresija.

Ovakvi preuslovi značajno sužavaju izbor dostupnih baza podataka. Razlog je taj što je veoma teško izazvati spontane mikroekspresije u veštačkim okolnostima. Ekman predlaže tri načina da se kreiraju situacije u kojima ispitanik može prikazati spontane mikroekspresije [13]:

- Zahtevati od ispitanika da lažu o tome šta su videli na video zapisu
- Konstruisati kriminalne scenarije
- Zahtevati od ispitanika da lažu o svom ličnom mišljenju

Motivacije ispitanika da uspešno lažu mogu biti raznovrsne:

- Osobe koje neuspešno lažu moraju da popune duge i dosadne formulare
- Osobe koje uspešno lažu biće dodatno plaćene
- Ispitanicima je rečeno da će procena njihovih performansi direktno uticati na njihove karijere

Prethodna uputstva svakako doprinose povećanju broja prikazanih mikroekspresija kod ispitanika, ali ipak stopa pojave spontanih mikroekspresija je generalno niska. Iz tog razloga i broj mikroekspresija koje baza podataka sadrži je relativno nizak (do par stotina). Baze podataka spontanih mikroekspresija su:

- CASME: Chinese Academy of Sciences Micro-Expressions [28]
- SMIC: Spontaneous Micro-expression Corpus [17]
- CASME II: Chinese Academy of Sciences Micro-Expression II [29]
- SAMM: Spontaneous Actions and Micro-Movements [9]
- CAS(ME)²: A Dataset of Spontaneous Macro-Expressions and Micro-Expressions [24]

¹Tako da je vremensko trajanje ubrzane makroekspresije slično trajanju mikroekspresije

Tačnost programa za detekciju mikroekspresija je testirana nad bazama CASME i CASME II.

4.1.1 CASME

Subjekti su snimani kamerom koja beleži 60 slika u sekundi. Od 1500 spontanih pokreta lica izabrano je 195 spontanih mikroekspresija. Ukupan broj ispitanika je 35 od čega 13 žena i 22 muškarca. Svim ispitanicima je rečeno da održe neutralan izraz lica tokom snimanja video zapisa.

Kako bi izazvali mikroekspresije, ispitanicima su prikazani video zapisi koji imaju visoku emocionalnu izražajnost. Ispitanicima je rečeno da će im za svaku emociju koju pokažu biti odbijeno 5 RMB od plate,² kao i da je uspešno prikrivanje emocija visoko povezano sa stopom uspeha u društvenom životu. Video zapisi CASME baze podataka podeljeni su u dve grupe:

- Grupa A: Ispitanici su snimani kamerom rezolucije 1280x720 piksela u prostori-riji koja je prirodno osvetljena.
- Grupa B: Ispitanici su snimani kamerom rezolucije 640x480 piksela u prostori-riji sa veštačkim LED osvetljenjem.

Sve mikroekspresije su morale da ispune jedan od dva uslova:

- Dužina trajanja mikroekspresije je kraća od 500ms.
- Dužina trajanja od početka do vrhunca mikroekspresije je kraća od 250ms.

Brzi pokreti koji ne nose sa sobom emociju nisu uzeti u obzir pri odabiru mikroekspresija. Primeri takvih pokreta su: pomeraj usne pri gutanju pljuvačke, širenje nozdrva pri udisaju, pomeraj obrva pri promeni pravca pogleda, itd.

4.1.2 CASME II

Ispitanici su snimani kamerom rezolucije 640x480 piksela koja beleži 200 slika u sekundi. Oblast lica je približno veličine 280x340 piksela i veća je nego u CASME bazi podataka (gde je oblast lica približno 150x190 piksela). Svi ispitanici su snimani u veštačkom osvetljenju.³ Od 3000 spontanih pokreta zabeleženo je 247 spontanih mikroekspresija na licima 26 učesnika. U eksperimentu je učestvovalo ukupno 35 ljudi i oni su podeljeni u dve grupe. Prva grupa, od 18 ispitanika, je imala uputstvo da zadrži neutralan izraz lica tokom snimanja video zapisa. Druga grupa, od 17 ispitanika, je imala uputstvo da, kada uoče da dolazi do promene izraza lica, probaju što pre da potisnu pokrete koje dovode do promene izraza lica. Ostale postavke eksperimenta su iste kao i u CASME bazi podataka.

²Približno oko 70 dinara.

³4 LED Lampe usmerene ka kisobranima za osvetljenje koje su usmerene ka licu.

4.2 Eksperimentalni rezultati

4.2.1 Način procene tačnosti detektora

Sve mikroekspresije u bazama podataka uglavnom imaju označene redne brojeve slika početaka i kraja mikroekspresija. Neka je taj interval, za neku mikroekspresiju, dat sa $[me_onset, me_offset]$.

Ako bi detektor uspešno označio poziciju koja pripada $[me_onset - \frac{k}{2}, me_offset + \frac{k}{2}]$ smatramo da je mikroekspresija uspešno pronađena i beležimo $me_offset - me_onset + 1$ slika kao tačno klasifikovane. Pri tom, broj k predstavlja radijus intervala mikroekspresije N i jednak je $\frac{N-1}{2}$, gde N predstavlja broj slika od TF to HF kao što je to opisano u [glavi 2](#). U slučaju da je detektor netačno označio poziciju ℓ , koja ne pripada nijednoj mikroekspresiji, tada sve slike u intervalu $[\ell - k, \ell + k]$, koje ne pripadaju nijednoj mikroekspresiji, označavamo kao potvrđene greške.

Ovakav način označavanja tačno i netačno klasifikovanih slika potiče iz [\[18\]](#) i [\[21\]](#). Uzimajući u obzir broj tačno klasifikovanih slika i broj potvrđenih grešaka, koji se dobijaju testiranjem nad bazom podataka mikroekspresija, definišemo pojmove koji će služiti kao merilo tačnosti detektora mikroekspresija.

Procenat tačno klasifikovanih slika mikroekspresija, TPR , računa se kao odnos broja slika koje je detektor tačno označio da pripadaju mikroekspresijama i broja svih slika koje pripadaju svim mikroekspresijama. TPR možemo tumačiti kao verovatnoću da detektor uspešno detektuje mikroekspresiju.

Procenat potvrđene greške, FPR , računa se kao odnos broja slika koje ne pripadaju mikroekspresijama, a koje je detektor označio da pripadaju, i ukupnog broja svih slika koje ne pripadaju nijednoj mikroekspresiji. FPR možemo tumačiti kao verovatnoću da detektor označi deo snimka koji nije mikroekspresija. Velika vrednost FPR nužno povlači i veću količinu vremena koje je potrebno uložiti u ručno analiziranje snimka u potrazi za mikroekspresijama.

Bitno je napomenuti da vrednosti FPR i TPR direktno zavise od parametra τ koji reguliše osetljivost detektora, kao što je to već opisano formulom [2.65](#). Za sve baze podataka procena tačnosti detekcije mikroekspresija je vršena za sve vrednosti τ počev od 0 do 1 sa korakom 0.05.

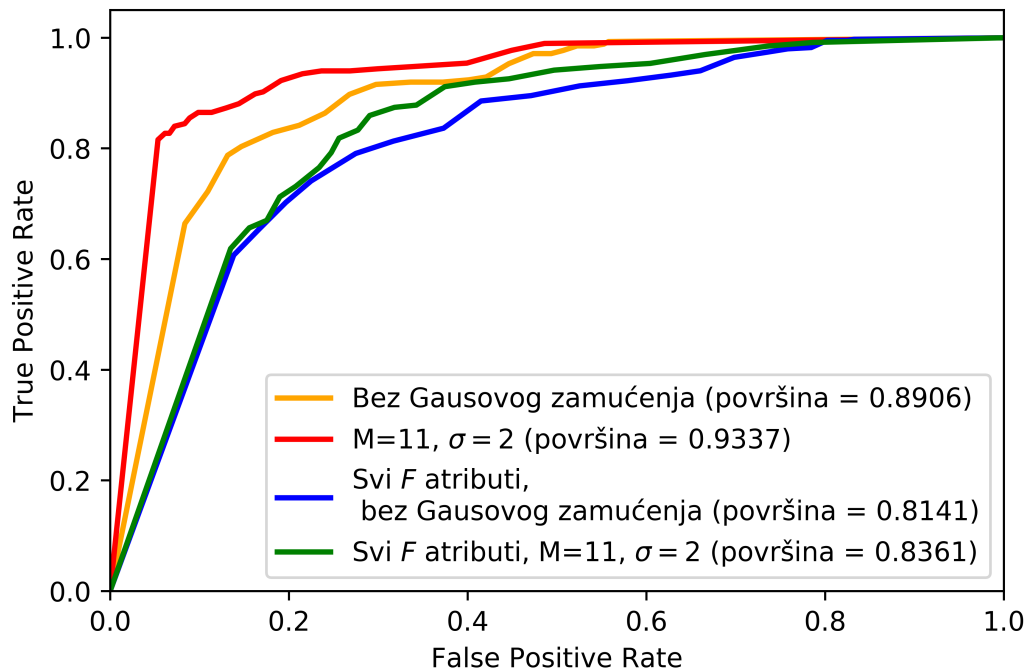
4.2.2 Rezultati za CASME II

Za CASME II bazu podataka odabran je interval mikroekspresije N u trajanju od 0.335 sekundi (67 slika). Za računanje vrednosti F atributa korišćene su uniformne vrednosti LBP atributa i svi histogrami su normirani. Korišćeni su LBP atributi populračnika 3 sa 8 suseda.

Za testiranje su korišćeni svi video zapisi mikroekspresija, u njihovom izvornom obliku, za koje je moguće izračunati C atribut vrhunca mikroekspresije. Njih ima ukupno 146, ukupnog trajanja oko 3.7 minuta. Prosečno trajanje video snimka je 1.56s. Video zapisi gde je vrhunac mikroekspresije u prvih ili poslednjih 0.335 sekundi nisu razmatrani, jer za vrhunce tih mikroekspresija nije moguće izračunati C atribut.⁴

Tačnost prethodno opisanog detektora opisana je grafikom obojenim narandžastom bojom na [slici 4.1](#), gde je na y osi prikazan procenat tačno klasifikovanih slika,

⁴Zapravo ako bi, pri računanju F atributa prvih k slika uzeli za TF prvu sliku video zapisa, uspeli bi da nađemo C attribute i za one slike koje se nalaze između 0.165s i 0.335s. Slično važi i za poslednjih k slika kada bi odabrali za HF poslednju sliku video zapisa. Ipak ovo nije optimalan način računanja F atributa.



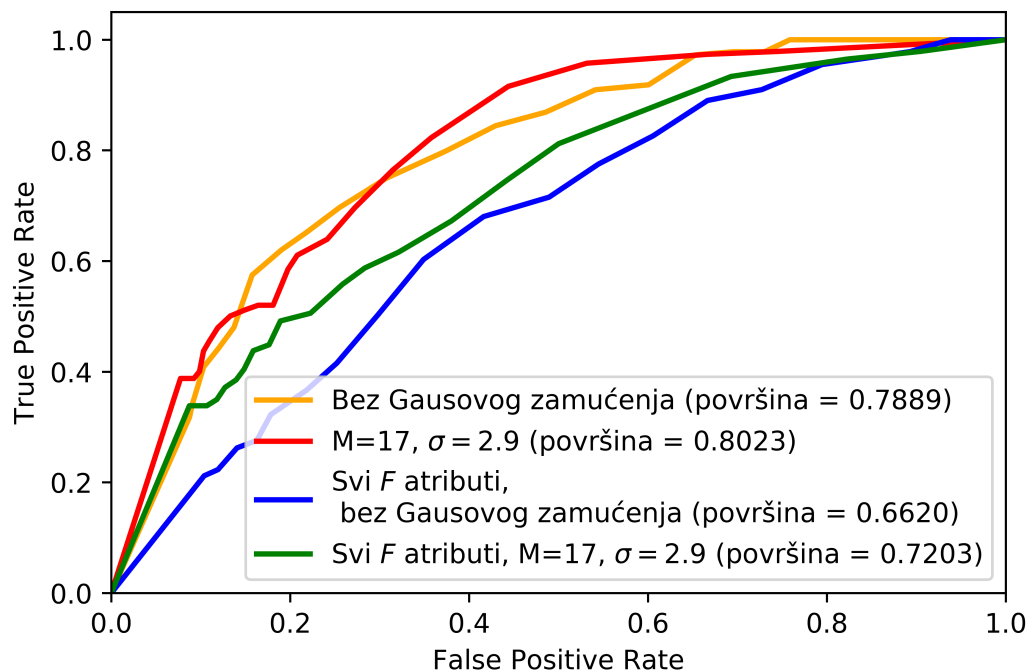
Slika 4.1: Tačnost detektora mikroekspresija pri testiranju CASME II baze. U slučaju primene Gausovog zamućenja, u koraku pretprocesiranja, vrši se konvolucija nad matricom $M \times M$ koja sadrži vrednosti Gausovog zvona standardne devijacije σ .

TPR , a na x osi procenat potvrđene greške, FPR . Površina ispod krive odgovara verovatnoći da detektor rangira nasumično odabranu sliku mikroekspresije više od nasumično odabrane slike koja ne pripada mikroekspresiji, i ona je u slučaju CASME II i prethodno pomenute konfiguracije jednaka 0.8906 što odgovara verovatnoći od 89.06%.

Primenom Gausovog zamućenja na sve slike video zapisa u koraku pretprocesiranja dobija se bolja tačnost detektora. Gausovo zamućenje, u ovom slučaju, postizemo konvolucijom slike video zapisa i matrice 11×11 koja sadrži vrednosti Gausovog zvona standardne devijacije 2. Koristeći Gausovo zamućenje, u procesu pretprocesiranja, dobijamo detektor čija tačnost je opisana crvenim grafikom na [slici 4.1](#). Površina ispod grafika je u tom slučaju 0.9337, što odgovara detekciji tačnosti 93.37%.

4.2.3 Rezultati za CASME

Testiranje nad CASME bazom podataka izvršeno je posebno za video zapise iz grupe A i video zapise iz grupe B. Interval mikroekspresije N postavljen je na 0.35 sekundi (21 slika). Za računanje vrednosti F atributa korišćene su uniformne vrednosti LBP atributa i svi histogrami su normirani. Korišćeni su LBP atributi poluprečnika 3 sa 8 suseda. Isto kao i pri testiranju CASME II baze, odbačeni su video zapisi za koje nije moguće izračunati C atribut vrhunca mikroekspresije. Ukupan broj video zapisa iz grupe A koji su korišćeni za testiranje je 83, ukupnog trajanja oko 4.28 minuta. Prosečno trajanje video zapisa iz grupe A, korišćenih za testiranje, je 3.09 sekundi. Ukupan broj video zapisa iz grupe B koji su korišćeni za testiranje je 81, ukupnog trajanja oko 4.94 minuta. Prosečno trajanje video zapisa iz grupe B, korišćenih za testiranje, je 3.66 sekundi.



Slika 4.2: Tačnost detektora mikroekspresija pri testiranju grupe A CASME baze. U slučaju primene Gausovog zamućenja, u koraku pretprocesiranja, vrši se konvolucija nad matricom $M \times M$ koja sadrži vrednosti Gausovog zvona standardne devijacije σ .

Grafik obojen narandžastom bojom na [slici 4.2](#) oslikava tačnost prethodno opisanog detektora nad video snimcima iz grupe A. Površina ispod te krive je 0.7889 što odgovara detekciji tačnosti 78.89%. Slično kao i u slučaju CASME II baze podataka, rezultati su bolji ako se koristi Gausovo zamućenje. Grafik obojen crvenom bojom na [slici 4.2](#) opisuje tačnost detektora sa Gausovim zamućenjem primenjenim nad video snimcima iz grupe A. Površina ispod takve krive je 0.8023 što odgovara detekciji tačnosti 80.23%.

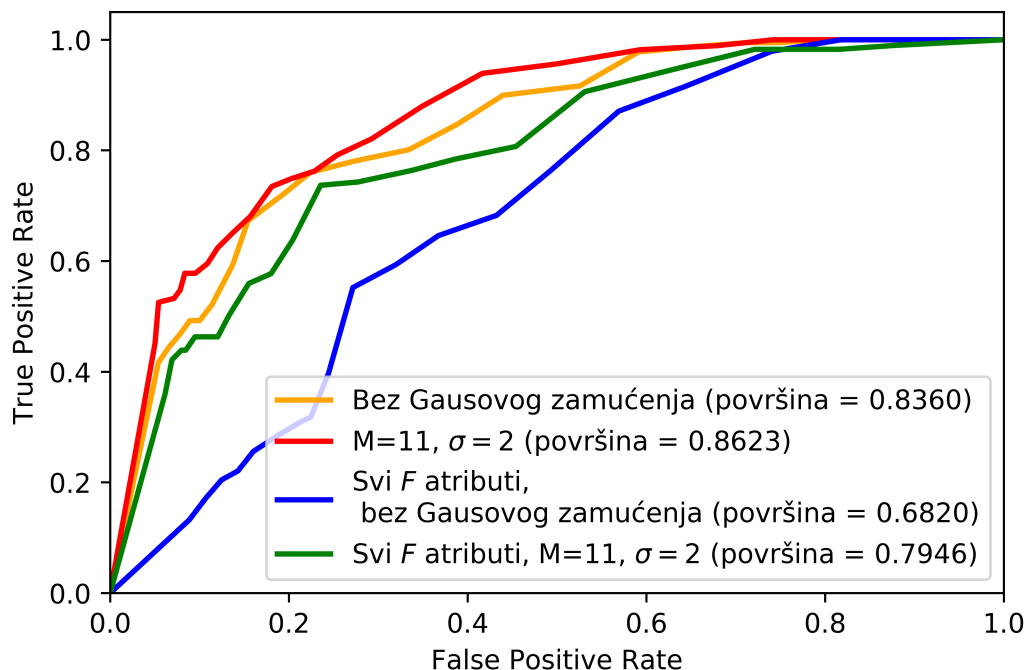
Gausovo zamućenje, primenjeno nad video snimcima grupe A, predstavlja konvoluciju slike video zapisa i matrice 17×17 koja sadrži vrednosti Gausovog zvona standardne devijacije 2.9. U ovom slučaju matrica sa vrednostima Gausovog zvona je nešto većih dimenzija jer je i rezolucija video zapisa iz grupe A veća nego rezolucija video zapisa iz grupe B ili CASME II baze.

Tačnost detektora bez Gausovog zamućenja, nad video snimcima iz grupe B opisana je narandžastim grafikom na [slici 4.3](#). Površina ispod krive je 0.8360 što odgovara detekciji tačnosti 83.60%. Grafik obojen crvenom bojom na [slici 4.3](#) opisuje tačnost detektora nad video snimcima iz grupe B nad kojima je korišćeno Gausovo zamućenje u koraku pretprocesiranja. Površina ispod takve krive je 0.8623 što odgovara detekciji tačnosti 86.23%. Gausovo zamućenje primenjeno nad slikama video zapisa iz grupe B isto je kao i u slučaju CASME II baze.

4.3 Poređenje rezultata

4.3.1 CASME II

Dobijeni rezultati su neznatno bolji od onih dobijenih u [18]. Kriva koja opisuje tačnost detektora, datog u [18], data je na [slici 4.4](#). Površina ispod crvene krive



Slika 4.3: Tačnost detektora mikroekspresija pri testiranju grupe B CASME baze. U slučaju primene Gausovog zamućenja, u koraku pretprocesiranja, vrši se konvolucija nad matricom $M \times M$ koja sadrži vrednosti Gausovog zvona standardne devijacije σ .

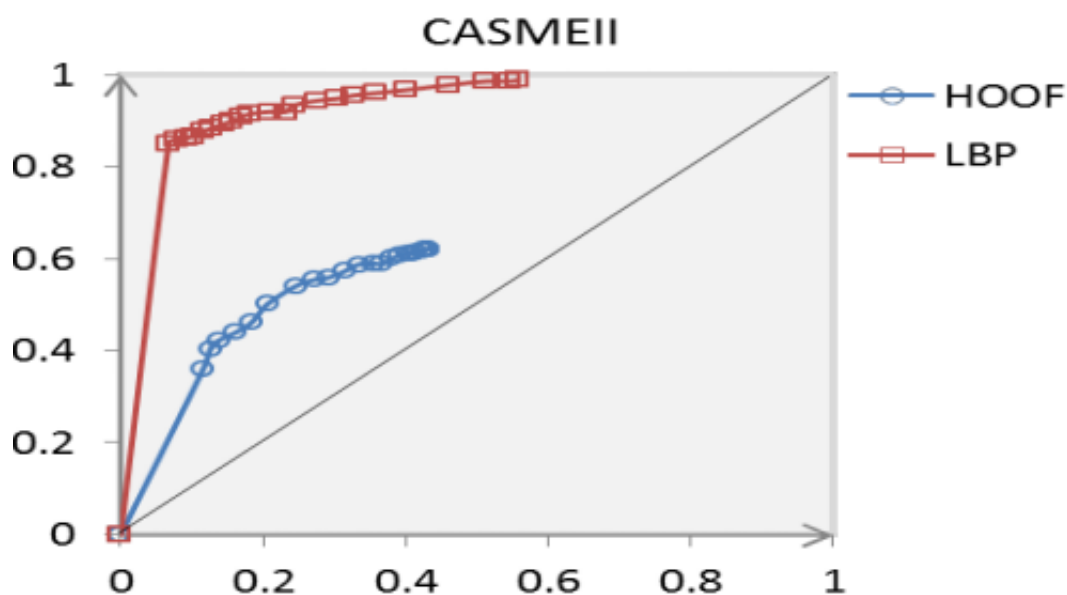
je 0.9298. Ipak, način na koji su rezultati dobijeni se razlikuju. U [18] testirani su svi, sem jednog video zapisa CASME II baze, iako nije rečeno kako su računali C attribute u slučaju da je mikroekspresija na samom početku ili na samom kraju video zapisa. U [21] to je rešeno tako što se računaju F atributi svih slika u slučaju da je mikroekspresija na početku ili na kraju video zapisa. To se postiže tako što se za TF uzme prva slika video zapisa pri računanju F atributa prvih k slika. Slično se radi i za poslednjih k slika i HF . Takav način računanja F atributa može dovesti do dodatne osetljivosti detektora na pokrete sa početka, odnosno kraja video zapisa.

Pri računanju F atributa svih slika, bez primene Gausovog zamućenja, postiže se tačnost od 81.41% kao što je prikazano grafikom plave boje na slici 4.1. Grafik obojen zelenom bojom, sa slike 4.1, opisuje tačnost detektora uz primenu Gausovog zamućenja u koraku pretprocesiranja, pri računanju F atributa svih slika.

Kao što se da primetiti, Gausovo zamućenje utiče pozitivno na tačnost, dok računanje F atributa svih slika može otežati detekciju onih mikroekspresija koje se ne nalaze na samom početku ili kraju video snimka. Zbog prethodnog, pri testiranju detektora opisanog u ovom radu, nije menjana konfiguracija testiranja u odnosu na lokaciju mikroekspresija, jer u suprotnom koristimo informaciju o lokaciji mikroekspresija pri testiranju. To može dovesti do značajnih odudaranja od pravog rezultata.

Još jedna bitna razlika jeste da su u [18] ključne tačke uglova oka i podnožja nosa nalazili ručno, na početku svakog video zapisa, a zatim koristeći Kanade-Lucas-Tomasi algoritam pratili te tačke kroz ostatak video zapisa. U ovom radu, detekcija ključnih tačaka se radi automatski za svaku sliku video zapisa.

Nije zgoreg pomenuti i trenutno najbolji sistem za detekciju mikroekspresija za CASME II bazu [10]. U okviru tog sistema mikroekspresija se posmatra iz dva dela: od početka do vrhunca mikroekspresije i od vrhunca do kraja mikroekspresije. Za



Slika 4.4: Tačnost detektora datog u [18].

svaku sliku video zapisa računali su varijansu signala dobijenog na osnovu informacija iz piksela oblasti oka. Na osnovu vrednosti varijanse uspeali su da dobiju informaciju o trenucima kada je došlo do pomeraja očiju ili treptaja oka i koristeći tu informaciju smanje broj loših klasifikacija. Kako se detekcija mikroekspresija radi iz dva dela, signal će za jednu mikroekspresiju imati dva lokalna ekstremuma. U slučaju kada je treptaj oka na početku ili na kraju mikroekspresije i dalje možemo da se oslonimo na drugi ekstremum signala koji se vremenski ne poklapa sa treptajem oka. Površina ispod krive koja opisuje tačnost takvog detektora je tada 0.9513, odnosno detekcija mikroekspresija ima tačnost od 95.13%.

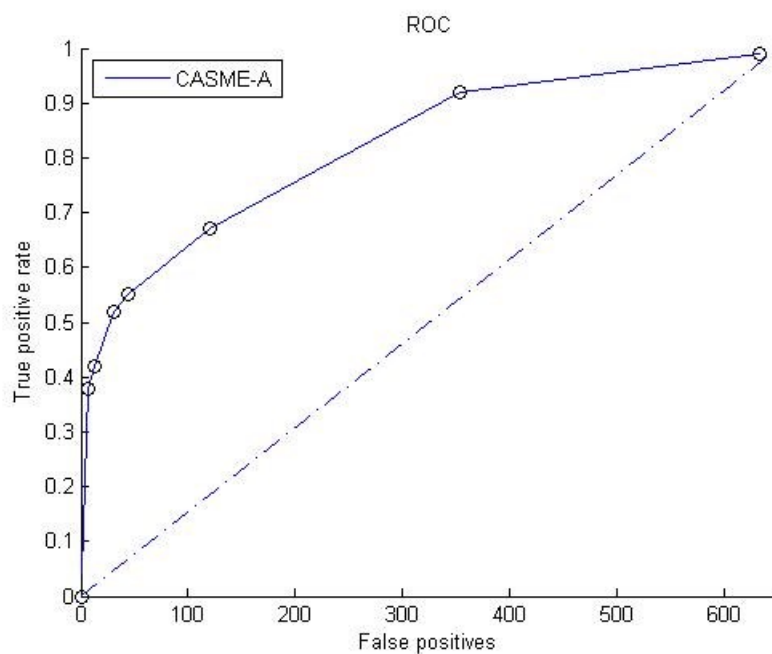
4.3.2 CASME

Pri analizi video snimaka iz grupe A, u [21], dobijena je tačnost od 82%. Kriva koja opisuje tačnost prilikom te analize data je na slici 4.5. Pri analizi snimaka iz grupe B, u [21], dobijena je tačnost od 90%. Kriva koja opisuje tačnost prilikom testiranja detektora nad snimcima iz grupe B data je na slici 4.6.

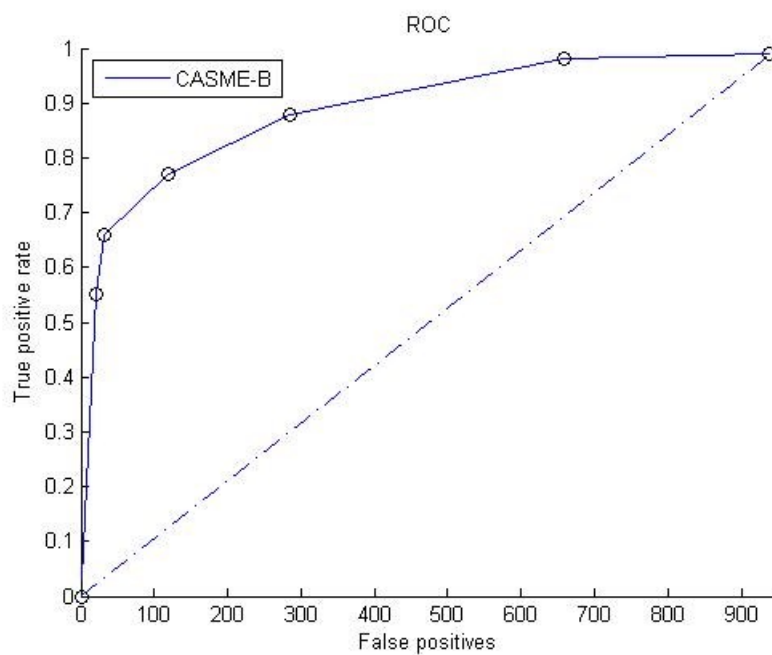
Iako je nad CASME bazom u ovom radu postignuta manja tačnost, bitno je napomenuti da u [21] treptaji oka, nezavisno o prisustvu mikroekspresija, nisu računati kao potvrđene greške. Takva konvencija može značajno povećati prikazanu tačnost detektora.

Još jedna bitna razlika jeste da su u [21] računali F atribut svih slika u slučaju da se mikroekspresija nalazi na samom početku ili kraju video snimka. Grafik obojen plavom bojom na slici 4.2, odnosno slici 4.3, opisuje tačnost detektora, pri analizi video snimaka grupe A, odnosno B, i računanju F atributa svih slika, bez primene Gausovog zamućenja. Grafik obojen zelenom bojom na slici 4.2, odnosno slici 4.3, opisuje tačnost detektora pri analizi video snimaka grupe A, odnosno B, i računanju F atributa svih slika, uz primenu Gausovog zamućenja u koraku pretprocesiranja.

Slično kao i u slučaju CASME II baze, u [21] ključne tačke su nalazili ručno na početku svakog video zapisa, a zatim ih pratili kroz ostatak snimka. U ovom radu detekcija mikroekspresija se vrši potpuno automatski.



Slika 4.5: Tačnost detektora datog u [21].



Slika 4.6: Tačnost detektora datog u [21].

4.4 Analiza rezultata

4.4.1 CASME II

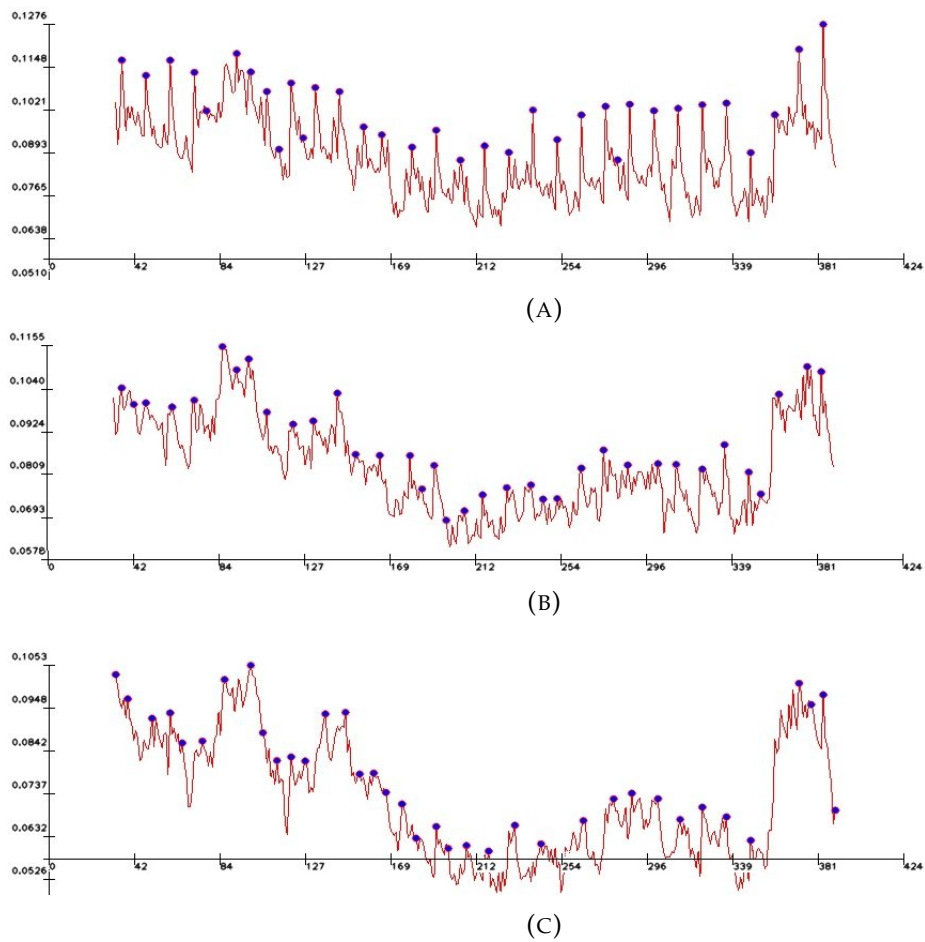
Šum igra značajnu ulogu pri netačno detektovanim mikroekspresijama na obodu lica. Zanimljivo je da korišćenjem histograma svih LBP vrednosti (ne samo uniformnih), dobijamo slične rezultate (89.32%), iako uniformne LBP vrednosti odgovaraju većini vrednosti LBP atributa lica (oko 90%) i potvrđeno je da korišćenjem uniformnih vrednosti smanjujemo uticaj šuma pri merenju razlike histograma. Kako kamera, kojom su snimani video zapisi CASME II baze, može beležiti maksimalno 200 slika u sekundi (i kako je poseban režim morao biti odabran kako bi se ta stopa dostigla) moguće je da određeni pikseli imaju izmenjene vrednosti zbog šuma koji potiče od kamere, ali ovo nije potvrđeno. Primenom varijante LBP atributa otpornih na šum nije postignuta veća tačnost. Ovo je potvrđeno korišćenjem NRLBP atributa [25] sa pragom nesigurnosti bita $t = 4$. Dobijena tačnost je 89%. Ovo nije neočekivano jer NRLBP atributi ne bi trebalo da dovedu do veće tačnosti u slučaju kada vrednosti piksela na slici puno odudaraju od svojih pravih vrednosti (odnosno kada je postavljena loša vrednost piksela). Kako je osvetljenje bilo skoro idealno pri snimanju subjekata iz CASME II baze, nije neočekivano da se korišćenjem NRLBP atributa postižu slični rezultati. Ipak, ne treba isključiti iz razmatranja varijante LBP atributa otpornih na šum u slučaju da se detektor koristi u prirodnom osvetljenju.

Bitno je primetiti da smanjivanjem vidljivosti detalja, odnosno visokofrekventnih komponenata slike, možemo smanjiti uticaj šuma pri analizi mikroekspresija. To možemo postići primenom Gausovog zamućenja.⁵ Savršeno vidljivi detalji nisu neophodni kako bi uočili mikroekspresiju. Zamislimo video snimak na kom dolazi do pokreta usne neke osobe. Da li bi posmatrač mogao da primeti da je došlo do pokreta usne ako bi Gausovo zamućenje bilo primenjeno na svaku sliku video snimka? Ako je stepen zamućenja (što u našem slučaju odgovara standardnoj devijaciji Gausovog zvona) relativno mali trebalo bi da je pokret primetan. Što je pokret sitniji to je potreban manji stepen zamućenja kako bi taj pokret bio vidljiv i na izmenjenom snimku.

Uticaj Gausovog zamućenja na F attribute, određenog snimka iz CASME II baze, može se videti na slikama 4.7a), 4.7b) i 4.7c). Na slici 4.7a) prikazani su F atributi koji se dobijaju u slučaju da Gausovo zamućenje nije primenjeno nad slikama video zapisa. Na slici 4.7b) prikazani su F atributi koji se dobijaju na osnovu slika nad kojima je primenjeno Gausovo zamućenje, implementirano kao konvolucija slike i matrice 5x5 koja sadrži vrednosti Gausovog zvona standardne devijacije 1.1. Na slici 4.7c) prikazani su F atributi koji se dobijaju na osnovu slika nad kojima je primenjeno Gausovo zamućenje, implementirano kao konvolucija slike i matrice 11x11 koja sadrži vrednosti Gausovog zvona standardne devijacije 2. Očigledno je smanjenje uticaja šuma, do te mere da je neke mikroekspresije, jačeg intenziteta, moguće uočiti direktno na osnovu vrednosti F atributa.

Za F attribute opisane slikama 4.7a), 4.7b) i 4.7c) mikroekspresija počinje od 69. slike i traje sve do 123. slike. Na slici 4.7a) nije očigledno da u tom intervalu F atributi imaju povišene vrednosti u odnosu na okolinu, dok je na slici 4.7c) to lako uočljivo. Povišene vrednosti F atributa na kraju video zapisa odgovaraju treptaju oka.

⁵Gausovo zamućenje predstavlja konvoluciju slike sa Gausovim zvonom $\frac{1}{2\sigma^2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}}$



Slika 4.7: Uticaj Gausovog zamućenja na vrednosti F atributa. Na slici su prikazani F atributi dobijeni: (a) bez Gausovog zamućenja (b) koristeći Gausovo zamućenje implementirano kao konvolucija nad matricom 5×5 (c) koristeći Gausovo zamućenje implementirano kao konvolucija nad matricom 11×11 .

4.4.2 CASME

Pri analizi video zapisa iz CASME baze nisu upotrebljene varijante LBP atributa otpornih na šum. Razlog tome je duga analiza video zapisa. Korišćenje svih vrednosti LBP atributa, ne samo uniformnih, ne dovodi do boljih rezultata u slučaju analize snimaka iz grupe A. U slučaju grupe B, rezultati su oko 2% bolji pri korišćenju uniformnih vrednosti nego u slučaju korišćenja svih vrednosti LBP atributa. To je možda i dobar pokazivač da ne treba odbaciti varijante LBP atributa otpornih na šum pri analizi video snimaka, ali kako je broj snimaka mali teško je izvesti takav zaključak sa sigurnošću.

Kao i u slučaju CASME II baze, korišćenje Gausovog zamućenja dovodi do bolje tačnosti. Razlog tome, kao što je već pokazano, jeste smanjenje uticaja šuma F atributa.

Grupa A sadrži video zapise koji su dosta dugi po trajanju (najduži video traje 11.72s!). Pri tom, u grupi A, postoje mikroekspresije koje traju duže od 0.5s, što je najduže pretpostavljeno trajanje mikroekspresije. Najduža mikroekspresija u grupi A traje skoro 1s. Prethodno navedena zapažanja mogu objasniti razliku između tačnosti dobijene analizom snimaka grupe A u odnosu na tačnost dobijenu analizom snimaka grupe B.

4.5 Performanse detektora

Kompletna analiza video zapisa vršena je na Intel Core i3-330M (2.13GHz) procesoru. Vreme potrebno za inicijalizaciju parametara je oko 5.74 sekundi. U koraku pretprocesiranja detekcija ključnih tačaka traje oko 49 milisekundi po slici, dok je za poziv `FaceAnalyser::AddNextFrame` potrebno dodatnih 9.8 milisekundi po slici. Za poziv metoda `NormalizedFace::addVideoFrame` i `NormalizedFace::writeNormalizedVideo` potrebno je, u proseku, oko 66 milisekundi po slici. Za potrebe samog analiziranja mikroekspresija potrebno je oko 34 milisekunde po slici.

Iako na i3-330M nije moguće vršiti analizu mikroekspresija u realnom vremenu, ono je svakako moguće. Algoritam je moguće paralelizovati, nakon čega je moguće iskoristiti računsku moć grafičkih kartica. Sami autori OpenFace biblioteke tvrde da je detekcija ključnih tačaka moguća u realnom vremenu. Kako je za problem pretprocesiranja i analize mikroekspresija potrebno jednako mnogo vremena možemo biti sigurni da je, uz dovoljno jak procesor i graficku karticu, detekcija mikroekspresija moguća u realnom vremenu. Ipak, kao što će to biti objašnjeno u [glavi 5](#), za tako nečim još uvek nema potrebe.

Glava 5

Pravci mogućih unapređenja

Iako neuronske mreže i druge metode mašinskog učenja postaju sve popularnije, trenutno je veoma teško je razviti sistem za detekciju mikroekspresija koji ima za osnovu neku od metoda mašinskog učenja. Jedan od problema je što postojeće baze podataka sadrže mali broj video zapisa mikroekspresija i ne sadrže snimke brzih pokreta lica koji nisu mikroekspresije. Brzi pokreti lica ne moraju biti mikroekspresije u slučaju da sa sobom ne nose emociju, kao što je slučaj sa pomeranjem usana pri gutanju pljuvačke, pomerajem obrva pri promeni pravca pogleda ili pomerajem nosa pri disanju. Video zapisi takvih pokreta, uz video zapise mikroekspresija, bi omogućili kreiranje robustnog sistema za detekciju mikroekspresija. Postojeće baze podataka ne sadrže takve video zapise pa je veoma teško, koristeći se metodama mašinskog učenja, na osnovu postojećih podataka, konstruisati robustan sistem za detekciju mikroekspresija. Postojeći sistemi za detekciju mikroekspresija, koji se ne oslanjaju na metode mašinskog učenja, takođe nisu robustni.

Glavni problem pri detekciji mikroekspresija jeste prepoznati koji treptaj oka je praćen emocijom a koji predstavlja čisto fiziološku reakciju. Trenutni sistemi uglavnom ili isključuju oblast očiju iz analize ili ne čine ništa po tom pitanju (što dovodi do toga da svaki treptaj oka može biti svrstan kao mikroekspresija). Problem se ne može jednostavno otkloniti, jer je oko 50% mikroekspresija praćeno treptajima oka [18]. Stoga je potrebno napraviti sistem koji, uz mikroekspresije, prepoznaje koji treptaji oka su značajni, a koji ne. Drugi način da se reši ovaj problem jeste da se konstruiše poseban sistem koji nezavisno analizira treptaje očiju i povratne informacije šalje sistemu za detekciju mikroekspresija.

Kako algoritmi za detekciju ključnih tačaka postaju tačniji i brži i kako broj baza podataka mikroekspresija raste otvaraju se nove mogućnosti u procesu detekcije mikroekspresija.

5.1 Metoda mašinskog učenja za proces detekcije mikroekspresija

Jedna takva metoda, koja postaje primenjiva tek sa porastom broja video zapisa mikroekspresija, bazira se na metodama mašinskog učenja.

Pronalaženje ključnih tačaka regularizovanim prosečnim pomerajem (objašnjenje metode je dato u [glavi 2](#)), pri procesu optimizacije, pronalazi i optimalne nerigidne parametre q .¹ Za potrebe detekcije mikroekspresija ne interesuju nas rigidni

¹Pogledati jednačinu [2.12](#)

parametri,² jer same pomeraje glave ne uzimamo u razmatranje pri analizi mikroekspresija.³ Na osnovu nerigidnih parametara q moguće je rekonstruisati, iako ne u potpunosti, kako samo lice tako i izraz lica osobe. Nerigidne parametre možemo zatim proslediti neuronskoj mreži kao ulazne parametre i na taj način istrenirati detektor mikroekspresija. Dati parametri su kvalitetan izbor ulaznih parametara, jer predstavljaju vektor koordinata u koordinatnom sistemu dobijenim analizom glavnih komponenti. Grubo rečeno, eliminisanjem redundantnosti iz informacija koje pružaju koordinate ključnih tačaka kreirali smo nerigidne parametre q .

Takođe što je detektor ključnih tačaka precizniji to je količina šuma sa kojom neuronska mreža treba da se izbori manja. Ipak, najbitnija stvar pri kreiranju ovakvog sistema jesu kvalitetni i brojni podaci. Povećanjem broja baza podataka mikroekspresija, kao i kvalitetnih primera brzih pokreta koji nisu mikroekspresije možemo očekivati i povećanje tačnosti (i popularnosti) sistema za detekciju mikroekspresija koji se baziraju na metodama mašinskog učenja.

5.2 Posebna analiza za treptaje oka

Kako su treptaji oka glavni krivac za veliki broj grešaka prvog tipa (lažnih pozitivna) nije zgoreg kreirati sistem koji posebno detektuje treptaje oka koji ne nose nikakav emocionalni značaj.

Ovakav sistem bi skoro sigurno bio veoma složen jer je učestalost treptaja značajna informacija pri merenju emocionalnog stresa osobe. Učestalost treptaja je problem sam za sebe i može značajno varirati od osobe do osobe. Odličan primer toga jeste da žene, koje unose kontraceptivna sredstva oralno, trepću 32% više od drugih žena. Prosečna osoba trepće oko 10 puta u minuti, ali kada čitamo učestalost treptanja može opasti i do 3-4 puta u minuti.⁴

Postoje i tri vrste treptaja oka: refleksni, spontani i voljni. Refleksni treptaji oka predstavljaju odgovor oka na spoljni stimulans, kao što je kontakt sa rožnjačom. Spontani treptaji oka, predstavljaju nekontrolisane treptaje koji ujedno nisu odgovor na spoljašnji uticaj. Voljni, odnosno kontrolisani treptaji oka, osim što koriste drugačiji skup mišića imaju i drugačiji intenzitet od spontanih treptaja. Za potrebe detekcije mikroekspresija od značaja su nam trenuci spontanih treptaja oka koji su praćeni određenim emocijama kratkog trajanja i intenziteta.

Iz prethodnog možemo zaključiti da je analiza treptaja očiju veoma složen problem. Pristupi rešavanja ovog problema mogu biti razni. Osim što možemo analizirati video zapis osobe i fokusirati se na oblast očiju, možemo analizirati i aktivnost električnih talasa u mozgu (EEG) u potrazi za određenim tipovima treptaja oka. Iako je krajnji cilj detektor mikroekspresija koji ne zahteva fizički kontakt sa ispitanikom, postoji mogućnost, dok se takav sistem ne implementira, korišćenja više različitih tehnologija u cilju što preciznije detekcije mikroekspresija.

²Faktor skaliranja, matrica rotacije i vektor translacije

³Ipak pomeraji glave nisu beskorisna informacija, oni mogu biti veoma korisni pri analizi govora tela

⁴Otud i umor očiju od čitanja.

5.3 Računanje rastojanja u svrhu detekcije mikroekspresija

Mikroekspresiju možemo smatrati i nesvesnom kontrakcijom mišića lica veoma blagog intenziteta. U tom slučaju, pri kontrakciji, dolazi do promene rastojanja između određenih ključnih tačaka, ma koliko ta promena bila mala. Sitne greške u proceni pozicija ključnih tačaka bi dovele do oscilacija u rastojanju. Ipak te oscilacije potiču iz šuma detektora ključnih tačaka, a ne od mikroekspresije.

Kako bi implementirali ovakav detektor neophodno je naučiti koje promene rastojanja potiču iz šuma, a koje iz mikroekspresije. Pri tom bi, uzimajući u obzir rastojanja između određenih ključnih tačaka, trebalo da je jednostavnije razgraničiti koji treptaj oka je čisto fiziološka reakcija, a koji je deo mikroekspresije. Takođe mogli bi detektovati i širenje zenice uz adekvatan detektor ključnih tačaka oka. I ovde važi da precizniji detektori ključnih tačaka povlače sa sobom manju količinu šuma sa kojom sistem treba da izađe na kraj, što bi trebalo da dovede do kvalitetnijih rezultata.

Glava 6

Zaključak

U ovom radu je pokazano kako veoma jednostavnom metodom, Gausovim zamućenjem, možemo povećati tačnost sistema za detekciju mikroekspresija. Dobijena tačnost sistema je bliska postojećim sistemima za detekciju mikroekspresija. Zanimljiv pravac razvoja, na koje upućuje Gausovo zamućenje, jeste aproksimacija delova lica glatkim površima. Nagla promena vrednosti u tačkama površi može odgovarati mikroekspresiji. Ipak i u ovom slučaju, kako bi razvrstali treptaje oka proračunih mikroekspresijom od onih koji to nisu, neophodna je veća količina podataka.

Iako sistemi za detekciju mikroekspresija imaju sve bolje performanske, realnost je da ovakav sistem nije primenljiv u svim slučajevima. Trenutni sistemi za detekciju mikroekspresija svakako mogu biti korisni u slučajevima kada podelimo video zapis ispitivanja na kratke delove koji sadrže po jedno pitanje i jedan odgovor (čak i to ne bi bilo loše podeliti u dva odvojena video zapisa). Zatim je neophodno dalje analizirati delove video zapisa, na koje upućuje detektor mikroekspresija, i potvrditi (ili opovrgnuti) postojanje mikroekspresije. Ovo je svakako veoma mali domen primene detektora mikroekspresija, ali duži video zapisi sa sobom donose i veću količinu grešaka, a samim tim i potrebnog vremena dodatne analize od strane ljudi.

Uzmimo za primer detektor koji ima stopu samo 1% da slike video zapisa, koje ne pripadaju mikroekspresiji, svrsta u klasu mikroekspresija (tj. da napravi grešku prvog reda). Pri ispitivanju subjekta u trajanju od 1h kamerom koja beleži samo 60 slika u sekundi imali bi dodatnih 36 sekundi materijala odnosno 2160 slika koje nisu deo mikroekspresija. Slike je zatim potrebno ručno analizirati i potvrditi koje slike su deo mikroekspresija a koje ne. Situacija na mestima kao što su granični prelazi i aerodromi je još gora uzimajući u obzir dužinu snimanja, kao i broj ljudi koji prođe kroz njih.

Iz prethodnog vidimo, da bi detektor mikroekspresija bio korišćen u masovnoj upotrebi, neophodno je da njegova tačnost bude veoma visoka, sa veoma niskom stopom greške prvog reda. Modularno rešavanje problema detekcije mikroekspresija, u nedostatku više podataka, deluje kao dobar način da se reše neki bitni problemi. Ipak, kako su se metode mašinskog učenja pokazale dominantnim u rešavanju raznih problema, za očekivati je da će sa nastankom novih i obimnijih baza, koje imaju za cilj treniranje takvog sistema, doći i do veće preciznosti sistema, koja će pak omogućiti masovnu primenu takvog sistema.

Reference

- [1] Timo Ahonen, Abdenour Hadid i Matti Pietikainen. „Face Description with Local Binary Patterns: Application to Face Recognition”. In: (2006). URL: <http://dip.lnmiit.ac.in/Projects/Paper16.pdf>.
- [2] Tadas Baltrušaitis, Peter Robinson i Louis-Philippe Morency. „Constrained Local Neural Fields for Robust Facial Landmark Detection in the Wild”. In: (2013). URL: <https://www.cl.cam.ac.uk/~tb346/pub/papers/iccv2013.pdf>.
- [3] Tadas Baltrušaitis, Peter Robinson i Louis-Philippe Morency. „OpenFace: an open source facial behavior analysis toolkit”. In: (2016). URL: <https://github.com/TadasBaltrušaitis/OpenFace>.
- [4] Tadas Baltrušaitis, Ntombikayise Banda i Peter Robinson. „Dimensional affect recognition using Continuous Conditional Random Fields”. In: (2013), pp. 3–5. URL: http://www.cl.cam.ac.uk/research/rainbow/projects/ccrf/files/fg_appendix.pdf.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006, pp. 435–443.
- [6] Miguel Á. Carreira-Perpiñán i Christopher K. I. Williams. „On the Number of Modes of a Gaussian Mixture”. In: (2003).
- [7] Yizong Cheng. „Mean shift, mode seeking, and clustering”. In: (1995). URL: http://home.ku.edu.tr/mehyilmaz/public_html/mean-shift/00400568.pdf.
- [8] T. F. Cootes i C. J. Taylor. „Active Shape Models — ‘Smart Snakes’”. In: (1992).
- [9] Adrian K. Davison et al. „SAMM: A Spontaneous Micro-Facial Movement Dataset”. In: (2018).
- [10] Carlos Duque et al. „Micro-Expression Spotting using the Riesz Pyramid”. In: (2018). URL: <https://hal.archives-ouvertes.fr/hal-01699355>.
- [11] Paul Ekman. *Emotions Revealed*. 2003, p. 32.
- [12] Paul Ekman i Wallace V. Friesen. „Nonverbal Leakage and Clues to Deception”. In: *Psychiatry - Journal for the study of interpersonal processes* (1969), 154–165. URL: <https://1ammce38pkj41n8xkpliocwe-wpengine.netdna-ssl.com/wp-content/uploads/2013/07/Nonverbal-Leakage-And-Clues-To-Deception.pdf>.
- [13] Mark G. Frank i Paul Ekman. „The ability to detect deceit generalizes across different types of high-stake lies”. In: (1997). URL: <http://web.missouri.edu/~segerti/1050H/docs/Ekman.pdf>.
- [14] Joseph C. Hager, Paul Ekman i Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. 1978.
- [15] E. Haggard i K. Isaacs. „Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy”. In: *New York: Appleton-Century-Crofts* (1966), 154–165.

- [16] Mary Katsikitis. *The Human Face: Measurement and Meaning*. 2003, pp. 99–119.
- [17] Xiaobai Li et al. „A Spontaneous Micro-expression Database: Inducement, collection and baseline”. In: (2013). URL: <http://tomas.pfister.fi/files/li2013microexpressions.pdf>.
- [18] Xiaobai Li et al. „Towards Reading Hidden Emotions: A Comparative Study of Spontaneous Micro-expression Spotting and Recognition Methods”. In: (Feb. 2017). URL: <https://arxiv.org/pdf/1511.00423.pdf>.
- [19] Iain Matthews i Simon Baker. „Active Appearance Models Revisited”. In: *International Journal of Computer Vision Volume 60, Issue 2* (2004). URL: https://people.eecs.berkeley.edu/~efros/courses/AP06/Papers/matthews_ijcv_2004.pdf.
- [20] Baback Moghaddam i Alex Pentland. „Probabilistic Visual Learning for Object Representation”. In: (1997).
- [21] Antti J. Moilanen, Guoying Zhao i Matti Pietikainen. „Spotting Rapid Facial Movements from Videos Using Appearance-Based Feature Difference Analysis”. In: (2014). URL: <https://www.researchgate.net/publication/283732576>.
- [22] Jorge Nocedal i Stephen J. Wright. *Numerical Optimization*. 2006, pp. 254–262.
- [23] Jorge Nocedal i Stephen J. Wright. *Numerical Optimization*. 2006, pp. 157–164.
- [24] Fangbing Qu et al. „CAS(ME)²: A Database for Spontaneous Macro-expression and Micro-expression Spotting and Recognition”. In: (2017).
- [25] Jianfeng Ren, Xudong Jiang i Junsong Yuan. „Noise-Resistant Local Binary Pattern With an Embedded Error-Correction Mechanism”. In: (2013).
- [26] Jason M. Saragih, Simon Lucey i Jeffrey F. Cohn. „Deformable Model Fitting by Regularized Landmark Mean-Shift”. In: (2011). URL: http://ci2cv.net/media/papers/2011_IJCV_Saragih.pdf.
- [27] William O. Straub. „A Brief Look at Gaussian Integrals”. In: (2009), pp. 4–5. URL: <http://www.weylmann.com/gaussian.pdf>.
- [28] Wen-Jing Yan et al. „CASME Database: A Dataset of Spontaneous Micro-Expressions Collected From Neutralized Faces”. In: (2013). URL: <http://fu.psych.ac.cn/CASME/CASME.pdf>.
- [29] Wen-Jing Yan et al. „CASME II: An Improved Spontaneous Micro-Expression Database and the Baseline Evaluation”. In: (2014).