



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET
Katedra za računarstvo



Detekcija umora kod vozača

Predmet: Intelligentni sistemi

Mentor: prof. dr Leonid Stoimenov

Student: Mila Rančić 1370

Niš, 2023

Sadržaj

Uvod.....	3
Metode za detektovanje umora	4
Posledice umora na performanse u vožnji.....	6
Praktični deo	8
Detektovanje lica, oka i treptanja	8
Određivanje parametara za detekciju umora	11
Računanje <i>ukupne zamorenosti</i>	13
Zaključak	15
Literatura	16

Uvod

U današnje vreme, skoro svaka porodica poseduje bar jedan automobil. Na ulicama je sve više vozila, pa se samim tim stvaraju sve veće gužve, a ljudi za volanom provode mnogo više vremena nego ranije. Zbog načina života koji vode, većina njih je konstantno umorna i pod stresom, što se odrazilo i na broj saobraćajnih nesreća koji je naglo porastao zadnjih godina.

Mnogim ljudima, noćna ili duga vožnja je u opisu posla (vozači kamiona, zdravstveni radnici...). Ovo su poslovi koji su bitni za društvo, a retko se obraća pažnja na to koliko su vozači odmorni. Umor kod vozača jedan je od glavnih krivaca nesreća. Umoran vozač nije pretnja samo za svoj život nego i za živote drugih učesnika u saobraćaju.

Kada je čovek pospan, može doći do kratkog stanja nesvesti, nazvanog mikrosnivanjem. U ovim slučajevima, vozač može i dalje imati otvorene oči, ali ne kontroliše pravilno svoje vozilo. Ukoliko vozač zadrema na par sekundi, recimo 4, prostim računanjem dobijeno je da ako automobil ide brzinom od 100 km / h, tada će automobil preći više od 100 metara bez njegove kontrole. S druge strane vozači ne mora da budu inicijatori nesreće, ali kada su umorni, nisu fokusirani i sporije reaguju na iznenadne situacije koje mogu da se jave na putu. Zbog toga je potreban sistem za merenje nivoa umora vozača i upozoravanje vozača kada se oseća pospano kako bi se izbegle nezgode.

Sistemi za detekciju umora mogu da koriste kamere, senzore za praćenje očiju, hardvere za praćenje vizuelnih znakova, gde se pospanost može otkriti kroz učestalost zevanja, učestalost treptanja očiju, pokrete pogleda, pokrete glave i izraza lica. Sistemi takođe mogu da nadgledaju ponašanje prilikom vožnje da bi primetili kada postoje nepravilni pokreti upravljača, korišćenje pedala i odstupanja u traku.

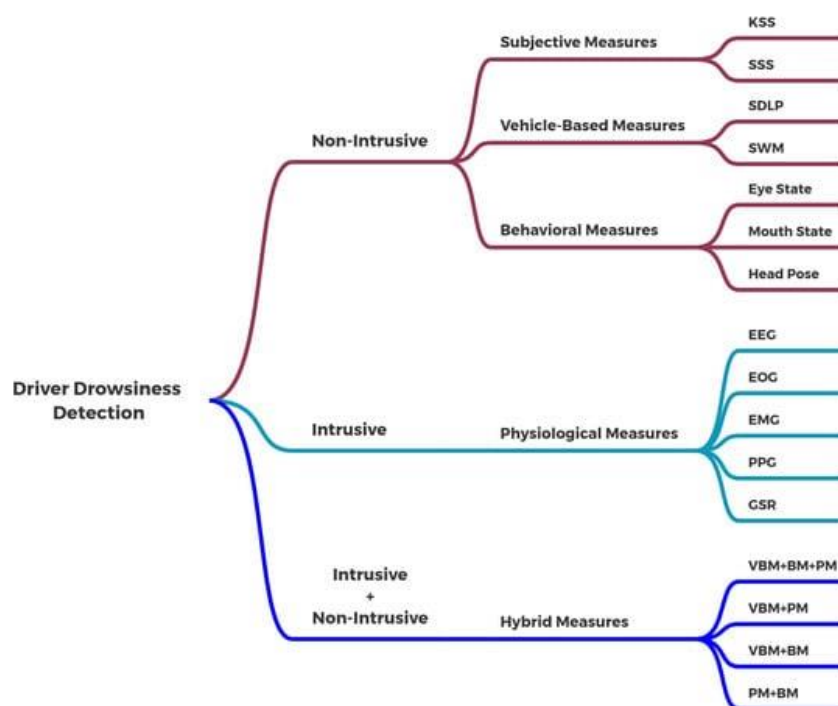
U ovom radu, obrađene su metode za detektovanje umora kod vozača. Postoji više načina da se detektuje umor. Jedan od njih je posmatranjem oka. Istraživači utvrđuju da se nivo pažnje normalne osobe u velikoj meri izražava stopom treptanja očiju. Ako se osoba oseća umorno ili pospano, menja se obrazac treptanja oka. Prosečno, čovek trepne 15 puta u minuti. Ukoliko je fokusiran na neku radnju, kao što je recimo vožnja automobila, treptanje se smanji na 7-10 treptaja u minuti. Takođe, bitna informacija je i koliko dugo treptaj traje. Prosečno vreme potrebno za izvršavanje pokreta zatvaranja i ponovnog otvaranja oka je oko 0,3 sekunde u minuti. Tako postaje jasno da se u zavisnosti od situacije u kojoj se nalazi osoba i njene usredsređenosti na situaciju menja brzina treptanja.

U okviru praktičnog dela, kreirana je aplikacija za detektovanje umora praćenjem učestalosti i dužine treptaja u minuti. Prvi korak je otkrivanje dela očiju, a zatim sledi detekcija treptanja. Koristeći dobijene podatke, utvrđena je frekvencija treptanja oka, kao i dužina trajanja treptaja. Ukoliko dobijene vrednosti ne pripadaju opsegu prosečnog nivoa, smatra se da je osoba umorna ili da joj je nivo pažnje manji, što svakako okida aktiviranje alarma.

Metode za detektovanje umora

Istraživanjima je ustanovljeno da se fiziološko stanje i ponašanje tokom vožnje drastično menja tokom određenog perioda pre nastupanja nesreće zbog umora. Na osnovu ovih promena urađeno je mnoštvo istraživanja i predloženo je mnogo metoda za detektovanje umora kod vozača. One se mogu podeliti u tri kategorije:

1. Intrusive,
2. Non-intrusive i
3. Hybrid



Slika 1: Metode detekcije umora kod vozača

Kao što vidimo sa slike 1, detektovanje na osnovu fiziološkog stanja/fizioloških informacija spada u Intrusive kategoriju. Ovde se analiziraju electroencephalogram (EEG), electrocardiogram (ECG), electrooculogram (EOG), electromyography (EMG) [5], respiratorni signali, otkucaji srca itd.

Subjektivne metode, zasnovane su na ponašanju vozača i vozila i ponašanja smatraju Non-intrusive. Podrazumevaju detektovanje stanja u kome se nalaze oči, detektovanje treptanja (što je tema ovog rada), pokreta glave, zevanja, praćenja pogleda, zatim, ukoliko se radi o vozilu detektovanje ugla točkova, volana, brzine, ubrzanja, gaženja linija na putu itd.

Hibridne metode su kombinacija dve ili više metoda.

Metode iz prve grupe spadaju u najpouzdanije zato što prate unutrašnje stanje u kome se nalazi telo vozača. One imaju najmanju stopu greške i najmanje su podložne spoljnim uticajima koji bi mogli da ugroze tačnost rezultata. Međutim, postojeće metode za detekciju na osnovu fiziološkog stanja su na neki način invazne i podrazumevaju postavljanje elektroda, ili sličnih stvari, na telo vozača, što može biti neprijatno, nelagodno, a u krajnjem slučaju može i smetati u vožnji. Pored toga,

spadaju u najskuplje, u smislu potrebne opreme. Zbog svega toga su i najmanje primenjene u odnosu na metode iz ostale dve grupe.

Što se tiče metoda koje rade detektovanje ponašanja vozača, one imaju tu prednost da se primenjuju u beskontaktnom maniru, bez ikakvog uticaja na vozača, u poređenju sa metodama iz prve grupe. Podrazumevaju korišćenje kamere za dobijanje slike lica vozača i za detektovanje pokreta glave, zbog čega spadaju u jeftinije metode po pitanju potrebne opreme. Koriste tehnike *računarskog vida (computer vision)* za otkrivanje umora. Loša strana ovih metoda je što se one primenjuju u kasnijoj fazi umora vozača, kada su očigledne promene na licu i očima, odnosno kada su promene u ponašanju već ušle u opasnu fazu po pitanju alarmantosti. Pored toga, ove tehnike su osetljive na spoljne faktore.

Metode za detektovanje umora na osnovu informacija o vozilu takođe spadaju u beskontaktne i primenljive su u tzv. "real time" sistemima, odnosno sistemima koji funkcionišu u realnom vremenu. Kada govorimo o ograničenjima kod njih, tu je pre svega cena potrebne opreme. Recimo za detektovanje skretanja sa trake na putu potrebno je da se pribavljaju slike puta i obrađuju u realnom vremenu, a to zahteva malo skuplji hardver. Treba obraditi i spoljne faktore poput znakova na putu, osvetljenja itd. Detektovanje volana, odnosno ugla za koliko je on okrenut, bi bilo jeftinija varijanta kod metoda iz ove grupe, ali je ona primenjiva samo u veoma specifičnim uslovima i kod određenog broja vozila.

Nakon pribavljanja velike količine podataka navedenim metodama, potrebno ih je dobro obraditi za bolje određivanje nivoa umora vozača. Metode obrade podataka implementiraju se pomoću matematičkih modela i mašinskog učenja. Implementacije zasnovane na mašinskom učenju podrazumevaju obuku na osnovu velike količine podataka o vožnji dobijenih iz laboratorije i sa puta, što je algoritam vođen podacima. Matematički modeli se odnose na kompleksne računice na osnovu parametara dobijenih u toku vožnje. Često se viđa i zajednička primena matematičkih modela i mašinskog učenja, što daje još bolje rezultate jer uzima najbolje iz oba načina, a može pokriti greške koje nastanu u jednom od njih.

Posledice umora na performanse u vožnji

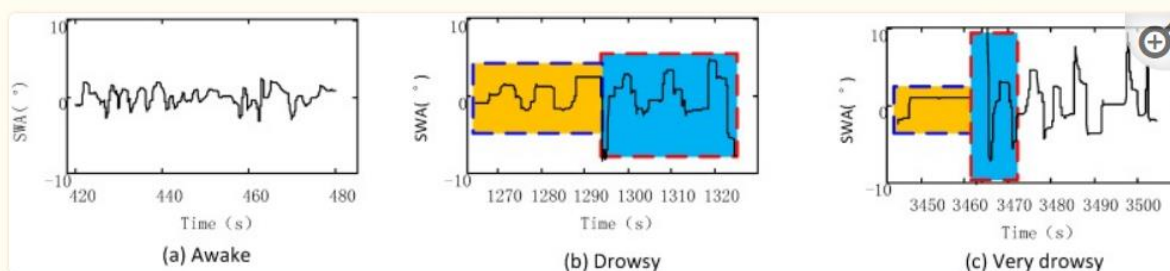
Umorni vozači imaju sporije vreme reakcije i pate od smanjene pažnje, svesti i sposobnosti da kontrolišu svoja vozila. Dokazano je da umorna vožnja može biti jednako opasna kao i vožnja u alkoholisanom stanju.

Prema ispitivanju <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5492517/#B14-sensors-17-01212>, vršen je eksperiment nad vozačem koji započinje vožnju u odmornom stanju, ali zbog dužine vožnje postaje sve umorniji. Korišćenjem metode za detektovanje umora na osnovu informacija o vozilu(uglova volana (SWA) i uglova skretanja (YA)), pomoću dijagrama, slikovito su prikazne promene performansi.

Na početku ispitivanja definisana su tri osnovna stanja: kad je vozač bio odmoran(budan), pospan i veoma pospan.

Stanje	Karakteristike
Budan	Glava ostaje uspravna, a izrazi lica su bogati. Pažljiv prema životnoj sredini. Oči se široko otvaraju i brzo trepću, a očne jabučice se aktivno kreću.
Pospan	Pažnja prema spoljašnjem svetu se smanjuje. Vozači prave gestove kao što su grebanje po licu, tresenje glavom, namigivanje, gutanje, uzdisanje, duboko disanje i zevanje. Oči imaju tendenciju da se zatvaraju, trepću polako sa manje aktivnosti očne jabučice.
Veoma pospan	Oči se dalje zatvaraju, a kapci postaju teži. Oči se duže zatvaraju. Vozač mož dremati, klimati glavom, nagnuti glavom, a zatim izgubiti sposobnost vožnje.

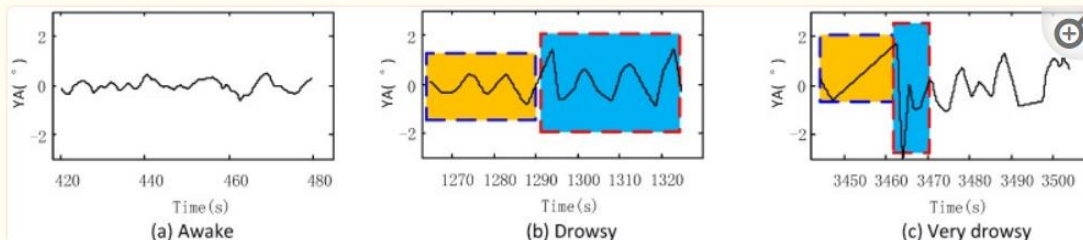
Tabela 1: Tri osnovna stanja vozača



Slika 2: Talasni oblici koji prate pomeranje uglova volana(SWA) tokom različitih nivoa umora vozača

Talasni oblici na slici 2 i slici 3 se koriste da vizuelno izraze kako nivoi umora vozača utiču na njegove sposobnosti prilikom vožnje. Na slici 2 prikazani su SWA talasni oblici na različitim nivoima zamora, iz čega možemo videti da kada je vozač budan, kao što je prikazano na slici 2a, on će često menjati uglove volana u malom opsegu. Kada je umoran, kao što je prikazano na slici 2b, frekvencija modifikacije je niska, što pokazuju talasni oblici u žutom bloku, dok amplituda modifikacije postaje veća većom brzinom, što je naznačeno u plavom bloku. Ako je vozač jako umoran, kao što je prikazano na slici 2c,

uglovi volana ostaju nepromenjeni tokom određenog vremenskog perioda, kao što je označeno žutim blokom, praćeno brzim fluktuacijama sa velikom amplitudom, kao što je prikazano plavim blokom.



Slika 3: Talasni oblici koji prate uglove skretanja(YA) tokom različitih nivoa umora vozača

Slika 3 prikazuje tipične YA talasne oblike pod različitim nivoima umora. Možemo videti da kada je vozač Budan, kao što je prikazano na slici 3a, talasni oblici često fluktuiraju u malom opsegu. Kada je umoran, kao na slici 3b, fluktuacije se usporavaju kao u žutom bloku, a promene se velikom brzinom povećavaju kao što je prikazano na plavom bloku. Kada je vozač izuzetno umoran, kao što je prikazano na slici 3c, talasni oblici postaju oštri dok je SWA nepromenjen, žuti blok, a zatim se javljaju brze i ogromne fluktuacije kako SWA modifikacija postaje brza i značajna, plavi blok.

Na ovom ispitivanju se vidi kako se uglovi volana i uklovi kako skreće vozilo menjaju u zavisnosti od zamora vozača. Kako njegov umor raste, SWA kod vozača opada u frekvenciji, povećava amplitudu velikom brzinom, pa čak i ostaje mirna tokom određenog vremenskog perioda. YA parametri takođe pokazuju karakteristike spore frekvencije, velike amplitude velikom brzinom.

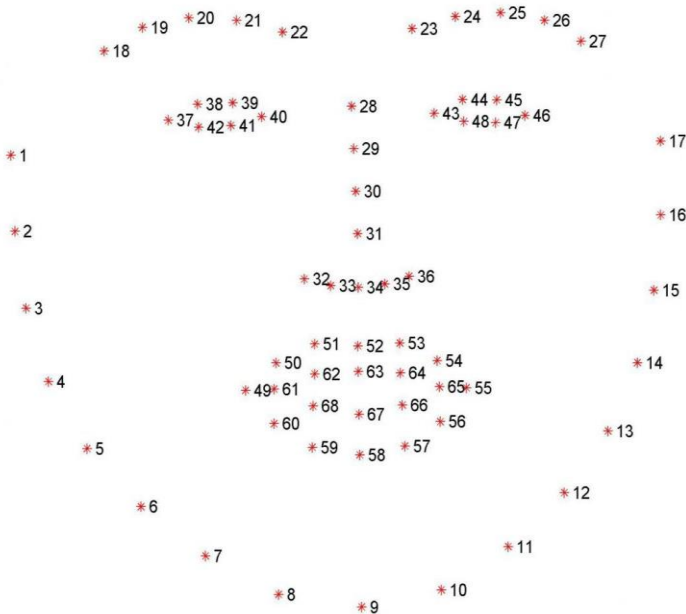
Praktični deo

Implementacija algoritma za detekciju umora može se podeliti u tri celine. U okviru prvog dela vrši se detektovanje lica, zatim očiju i treptanja na modelu lica sa slike koja se dobija sa kamere. Drugi deo bavi se određivanjem parametara na osnovu kojih se detektuje umor, a treći računanjem *ukupne zamorenosti* kojim se određuje nivo umora i procenjuje da li je situacija alarmanta ili ne.

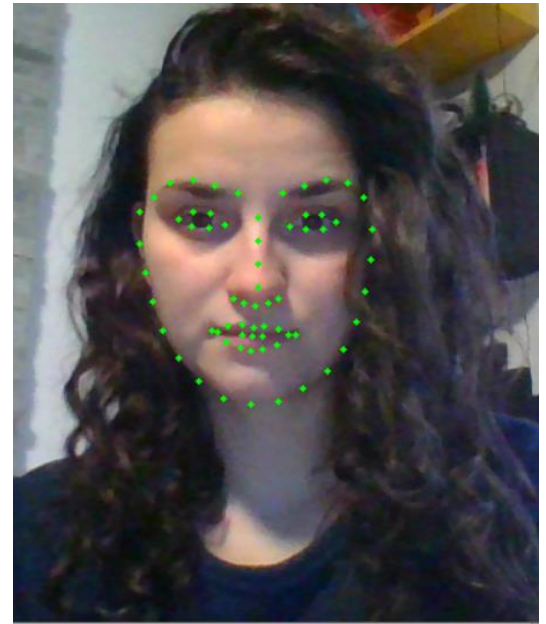
Detektovanje lica, oka i treptanja

Pre svega postavljena je kamera koja će biti zadužena za generisanje videa koji će se analizirati. Kamera je podešena na takav način da je na raspolaganju kompletna slika lica. Vrši se analiza svakog frejma videa koji se dobija sa kamere. Za otkrivanje lica i očiju korišćena je **dlib** biblioteka, odnosno unapred obučeni detektor lica zasnovan na modifikaciji *Histogram of oriented gradient* u kombinaciji sa linearnim *Support Vector Machines(SVM)* za klasifikaciju.

Slika 1 je primer dlib modela sa 68 tačaka. Ponekad nam nije potrebno svih 68 karakteristika, a ponekad nam treba i više od 68. Potrebno je prilagoditi te tačke zahtevima. Ovaj unapred obučeni detektor lica unutar dlib biblioteke koristi se za dobijanje pozicije (x,y koordinata) koje se preslikavaju na konture lica na slici lica sa kamere.



Slika 4: dlib model korišćen za detekciju lica sa kamere



Slika 5: Preslikavanje tačaka na slici lica sa kamere

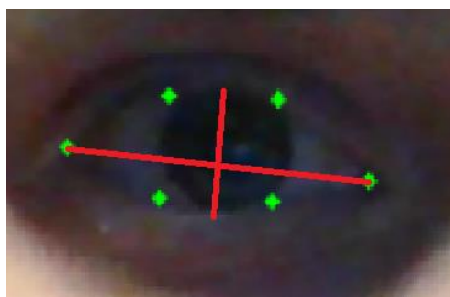
Svaka od ovih tački označena je rednim brojem kako bi moglo da se pristupi njenim koordinatama nakon što je izvršena detekcija lica. Ako je uspešno izvršena, rezultat ove početne faze bi trebalo da bude slika na kojoj su detektovane oči pomoću koordinata koje ih određuju.

Drugi deo je detekotvanje treptanja oka. Detektovanje treptanja oka je verovatno i najbitniji deo ovog algoritma, zato što učestvuje u obe komponente iz kojih se dobija kranji rezultat, a to su :

1. Detektovanje učestalosti(frekvencije, intervala) treptanja
2. Detektovanje koliko brzo se izvrši jedno treptanje oka - dužina treptanja

U narednim paususima biće detaljno objašnjeno kako ove komponente učestvuju u kreiranju konačnog rezultata. Za sada ćemo samo reći da je jako bitno uzeti u obzir oba ova problema pri detekciji umora.

Da bismo otkrili treptanje, prvo moramo da odredimo kada se oči smatraju otvorenim, a kada zatvorenim. To je urađeno određivanjem odnosa dužine horizontalne linije po sredini koja spaja krajnju levu i krajnju desnu tačku oka i vertikalne linije po sredini oka koja spaja središnje tačke gornjeg i donjeg kapka. Dužina se računa određivanjem Euklidske distance.



Slika 4: Horizontalna i vertikalna linija oka

Osnovni razlog zbog koga je izabrana baš ova metoda računanja koliko je oko otvoreno je pouzdanost - pokazala je najbolje rezultate prilikom testiranja na velikom broju uzoraka. Drugi razlog, skoro podjednako bitan, je taj što će dati tačne rezultate nezavisno od toga koliko je lice vozača udaljeno od kamere. Kada bi se, recimo, merila samo distanca između gornjeg i donjeg kapka, ona bi se smanjlila svaki put kada bi se vozač udaljio od kamere, odnosno povećala kada bi se približio. Na taj način bi dobijeni rezultati bili nevalidni.

Da bismo otkrili odnos dužina ove dve linije, uzimamo mapirane koordinate sa slike lica. Sa modela se vidi koju oznaku, odnosno redni broj, imaju koordinate koju sa nam od interesa. Zatim se pribavljaju podaci o vrednostima x i y koordinata za te tačke, na osnovu kojih se računa dužina horizontalne linije. Što se veritikalne tiče, računa se na sličan način, samo što se tačke između kojih se traži razdaljina dobijaju na sledeći način. Prva tačka se nalazi kao sredina između dve tačke gornjeg dela oka, a druga kao sredina između dve tačke donjeg dela oka.

```
def calculate_eye(eye_points, facial_landmarks):
    left_point = (facial_landmarks.part(eye_points[0]).x, facial_landmarks.part(eye_points[0]).y)
    right_point = (facial_landmarks.part(eye_points[3]).x, facial_landmarks.part(eye_points[3]).y)
    center_top = midpoint(facial_landmarks.part(eye_points[1]), facial_landmarks.part(eye_points[2]))
    center_bottom = midpoint(facial_landmarks.part(eye_points[5]), facial_landmarks.part(eye_points[4]))

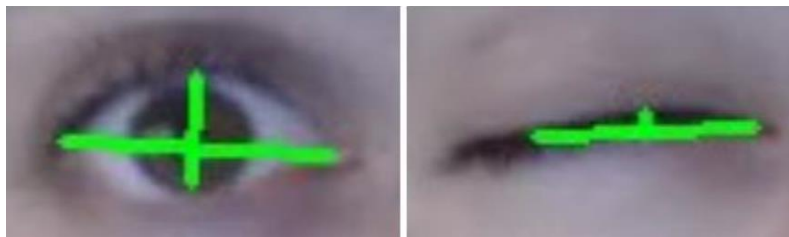
    hor_line_length = hypot((left_point[0] - right_point[0]), (left_point[1] - right_point[1]))
    ver_line_length: float = hypot((center_top[0] - center_bottom[0]), (center_top[1] - center_bottom[1]))

    ratio = hor_line_length / ver_line_length

    return ratio
```

Slika 5: Funkcija u okviru koje se računa eye aspect ratio

Na slici 7, prikazano je kako se određuju dužine linija i kako se pomoću njih računa odnos, eye aspect ratio (EAR). Ovaj odnos se koristi za upoređivanje sa unapred postavljenim pragom - ako je dobijena vrednost veća od vrednosti praga oko se smatra zatvorenim, a u suprotnom otvorenim. Vrednost za prag se dobija na osnovu vršenja eksperimenata na velikom broju uzoraka i finim podešavanjem vrednosti u cilju dobijanja što realnijih rezultata.



Slika 6: Odnos dužina linija kod otvorenog i zatvorenog oka

Ono što još treba napomenuti je da su prilikom računanja u obzir uzeta oba oka. To znači da je odnos dužina linija koje su pomenute urađen i za levo i za desno oko i određen je njihov zbir. Konačna vrednost se dobija tako što se taj zbir podeli sa dva što predstavlja srednju vrednost. Jednom kada se utvrdi da li su oči otvorene ili ne, može se detektovati treptanje. To dalje omogućava određivanje učestalosti treptanja.

Što se tiče određivanja vremena za koje se jedan treptaj izvrši, ono se određuje brojanjem frejmova. U tu svrhu koristi se još jedna vrednost, odnosno još jedna vrsta praga, pomoću koje se detektuje kada treptanje počinje. Ova vrednost dobijena je na osnovu prethodno određenog praga, takođe eksperimentisanjem na velikom broju uzorka. Više o tome videćemo u delovima koji slede.

Određivanje parametara za detekciju umora

Nakon detektovanja lica, očiju i treptanja, vrši se računanje svih potrebnih parametara za određivanje nivoa umora. Potrebno je odrediti dve stvari: učestalost, odnosno frekvenciju treptanja i koliko brzo se izvrši jedno treptanje oka. Oko može da se nađe u tri stanja.

1. Otvoreno
2. Treptanje započeto i
3. Zatvoreno

```
for face in faces:

    landmarks = predictor(gray, face)
    # facePoints2(frame, landmarks)

    left_eye = calculate_eye([36, 37, 38, 39, 40, 41], landmarks)
    right_eye = calculate_eye([42, 43, 44, 45, 46, 47], landmarks)
    blinking_ratio = (left_eye + right_eye) / 2

    if blinking_ratio > start_blinking_threshold: # 1st STATE - BLINKING HAS STARTED
        blinking_duration_frame_count += 1

        if blinking_ratio > blinking_threshold: # 2nd STATE - BLINKED
            eyes_were_closed = True
            if blinking_duration_frame_count > blinking_duration_frame_count_threshold:
                # print(blinking_frame_count)
                blinking_duration_alert_points += (blinking_duration_frame_count * blinking_duration_frame_count)
                total_alert_points += blinking_duration_alert_points
            else: # 3rd STATE BEGINS - EYES ARE OPENNED AGAIN
                if blinking_duration_frame_count > blinking_duration_frame_count_threshold:
                    # print(blinking_frame_count)
                    blinking_duration_alert_points += (blinking_duration_frame_count * blinking_duration_frame_count)
                    # print("blinking_duration_alert_points", blinking_duration_alert_points)
                    total_alert_points += blinking_duration_alert_points
                    blinking_duration_alert_points = 0
                    blinking_duration_frame_count = 0
```

Slika 7: Računanje *ukupne zamorenosti*

Na slici 9 prikazan je deo koda koji računa *ukupnu zamorenost*. Tokom svakog frejma vrši se provera. *Treptanje započeto* predstavlja stanje kada se oko polako zatvara i *ear* je veće od postavljenog praga. Oči su polu zatvorene, što, ukoliko traje predugo, može značiti da je vozač umoran. Zbog toga se postavlja parametar koji će da računa koliko dugo su oči bile u ovom stanju. Stanje *Zatvoreno* na neki način predstavlja podskup stanja *Treptanje započeto*, jer ako je zatvoreno, podrazumeva se i da je započelo. Zbog toga se u kodu stanje *Zatvoreno* proverava u okviru dela petlje za stanje *Treptanje započeto*. U zavisnosti od toga u kome se stanju nalaze oči vozača izvršava se deo koda, a krajnji cilj je računanje *ukupne zamorenosti*, zbir vrednosti dobijene računanjem zamorenosti u slučaju učestalog i predugog treptanja, na osnovu kojih se detektuje da li je umoran i otkriva koliko je situacija alarmantna.

Što se tiče logike koje se izvršava u zavisnosti od stanja u kome se nalaze oči, za prva dva stanja ona je krajnje jednostavna. Ako se oči nalaze u stanju *Treptanje započeto*, brojač frejmova

potrebnih da se izvrši jedno treptanje počinje da broji. Ovaj brojač se inkrementira svakog narednog frejma dok se ne detektuje da je oko opet otvoreno. Stanje *Zatvoreno* znači da je oko zatvoreno još više nego u prethodnom stanju. Zbog toga se u njemu postavlja flag da je *oko bilo zatvoreno*. Gotovo celokupna računica se odvija u trenutku kada se detektuje da je oko ponovo otvoreno, tj u stanju *oko je otvoreno*.

Ovde se, pre svega, vrednost brojača frejmova potrebnih da se izvrši treptanje upoređuje sa unapred postavljenim pragom. Prag je ovog puta određen na osnovu informacija sakupljenih o vremenu za koje prosečan čovek trepne jednom. Treptanje bi trebalo da traje 200-300 ms, dok u pospanom stanju ono može trajati i do 800-900 ms, nekada i više. Obzirom da jedan frejm traje nešto više od 30ms(obično oko 33ms), granica, odnosno prag je postavljen na 13, što je u prevodu oko 400ms. Ako je vrednost brojača veća od praga, računa *ukupna zamorenost*, a u zavisnosti od te vrednosti, određuje se nivo alarmantnosti(u sledećem delu ćemo videti i na koji način).

```
if eyes_were_closed:
    eyes_were_closed = False
    # Eyes were closed which means that blinking has happened,
    # so we need to calculate blinking frequency
    current_blinking_time = time.time()
    if last_blinking_time != 0: # to handle 0th case
        blinking_frequency = time.time() - last_blinking_time

    last_blinking_time = current_blinking_time
    if blinking_frequency < frequently_blinking_rate_time_threshold:
        frequently_blinking_critical_ratio = frequently_blinking_rate_time_threshold / blinking_frequency
        frequently_blinking_alert_points += frequently_blinking_critical_ratio
        frequently_blinking_frame_count += 1
    else:
        # Blink rate is in the 'normal' range again, reset everything
        if frequently_blinking_frame_count > 5:
            frequently_blinking_alert_points *= frequently_blinking_frame_count
            print("frequently_blinking_alert_points: ", frequently_blinking_alert_points)
            total_alert_points += (frequently_blinking_alert_points * 2)
            frequently_blinking_frame_count = 0
            frequently_blinking_alert_points = 0
```

Slika 8: Deo koda koji računa koliko je često vozač trepnuo

Nakon što je završeno detektovanje dužine jednog treptanja, prelazi se na deo koji se odnosi na učestalost, tj frekvenciju, treptanja. Na slici 10 prikazan je deo koda koji računa koliko je često vozač trepnuo. Na početku se proverava da li se dogodilo treptanje, na osnovu flag-a, koji je pomenut u prethodnom pasusu. Ukoliko je flag postavljen, znači da je oko bilo zatvoreno ranije i da je sada opet otvoreno, čime je i treptanje završeno, pa se računa frekvencija treptanja. Za ovo se koristi promenljiva u kojoj se pamti vreme kada je prethodno treptanje izvršeno. Ona se oduzme od promenljive koja predstavlja trenutno vreme čime se dobije interval između dva treptanja. Zatim se vrši zamena, trenutno vreme upisuje se u promenljivu u kojoj se pamti vreme kada je prethodno treptanje izvršeno kako bi ona mogla da posluži za sledeću računicu, odnosno određivanje narednog intervala. Kao što smo to već videli nekoliko puta, naredni korak je upoređivanje dobijene vrednosti intervala sa unapred postavljenim pragom. Vrednost praga je kao i kod prethodnog slučaja dobijena

na osnovnu prikupljanja informacija o učestalosti treptanja kod prosečne osobe, ali i na osnovu nekih sprovedenih eksperimenata nad uzorcima. Kod odrasle osobe, u normalnim uslovima, interval između dva treptanja može biti od 2 do 10 sekundi, dok se recimo kada je osoba fokusirana na nešto može povećati i do 15 ili 20 sekundi. Što se tiče problema detektovanja umora, što je manji interval između dva treptaja(vozač češće trepće), to je veća verovatnoća da je i umorniji, što znači da su oni obrnuto proporcionalni. Zbog toga, ako je dobijeni interval manji od praga, to je znak da bi situacija mogla biti alarmantna, pa se računa *ukupna zamorenost*.

Pored određivanja intervala, ovde se vodi računa o još jednoj jako bitnoj stvari. To je broj sukcesivnih frejmova u kojima je interval treptanja manji od "dozvoljenog", što na neki način predstavlja i broj uzastopnih bržih treptanja. Kažem da je ovaj podatak jako bitan iz dva razloga. Prvi je taj da, što je ovaj broj veći znači da vozač duže vreme trepće češće nego što je to uobičajeno, što je u velikom broju slučajeva znak da se nivo umora povećava, te je stoga i situacija je alarmantnija. Drugi je otklanjanje greške. To se odnosi pre svega na situacije kada vozač može da neki kraći period ima povećani interval treptanja iz nekog drugog razloga (čestica prašine upala u oko, svrab zbog nošenja naočara za vid, sunce itd). Tada vozač može da tokom kraćeg perioda trepće često, nakon čega će nastaviti sa treptanjem u intervalima u granicama normale. Podatak o broju sukcesivnih frejmova tokom kojih vozač češće trepće se u aplikaciji čuva u posebnom brojaču namenjenom za tu ulogu. Sa druge strane, ako je dobijeni interval veći od praga, to znači da je učestalost, tj frekvencija, treptanja opet u granicama normale. Tada se pristupa sabiranju poena koji se odnose na deo za učestalost treptanja, restartovanju brojača i promenljivih za poene itd.

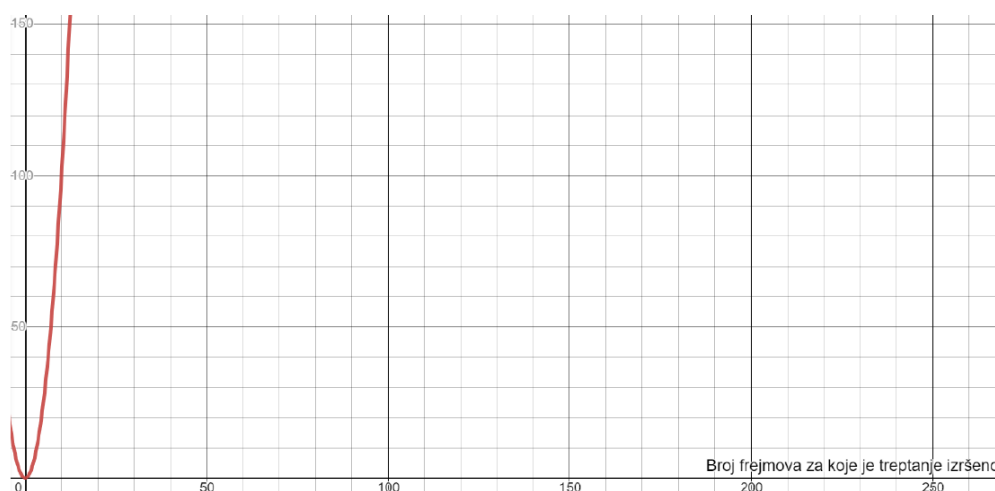
Računanje *ukupne zamorenosti*

Treći deo implementacije algoritma za detektovanje umora vozača odnosi se na procenjivanje koliko je, ili nije, vozač umoran. Tokom jednog frejma procenjuju se parametri, učestalost treptanja i vreme trajanja jednog treptanja, o kojima je bilo reči i na osnovu njih se dobija *ukupna zamorenost* tokom jednog frejma. Postupak se ponavlja u svakom narednom frejmu, pri čemu se već dobijena vrednost za *ukupnu zamorenost* uvećava za nove vrednosti. Kada zbir pređe određeni prag(vrednost nakon koje se smatra da je umoran), vozač će, putem zvučnog signala i poruke biti obavešten da je sistem ustanovio da je umoran i da mu je potreban odmor, ***You need to rest! Take a break!***.



Slika 9: Poruka praćena zvučnim signalom

Već je pomenut način dobijanja *ukupne zamorenosti* tokom jednog frejma, ali ono što nije obrađeno je koliko se koja komponenta vrednuje. Kao što je i logično, nije isto da li su oči vozača neko vreme bile u stanju *Treptanje započeto* ili *Zatvoreno* ili koliko se učestalo dešavala radnja treptanja. Počecemo od sabirka koji je vezan za vreme trajanje jednog treptanja. Bodovi za njega se računaju na osnovu broja frejmova potrebnih da vozač trepne. Što je veći broj frejmova, više se i poena dodaje na ukupan zbir, ali ne u odnosu 1 na 1. Već smo objasnili da što više frejmova jedno treptanje traje situacija je alarmantnija. Međutim, funkcija koja opisuje odnos ova dva faktora ne sme ni u kom slučaju biti linearna, jer nivo alarmantnosti mora da raste mnogo više u odnosu na samo povećanje broja frejmova. Na taj način možemo na vreme da otkrijemo da je vozač zadremao za volanom (ne otvara oči više od 2,3 sekunde) i da ga istog trenutka alarmiramo ne bi li se probudio. Zbog toga je za ovu svrhu izabrana kvadratna funkcija, gde X osa predstavlja broj frejmova za koje je treptanje izvršeno, a Y osa broj poena alarmantnosti:



Slika 10: Kvadratna funkcija za određivanje poena alarmantnosti na osnovu broja frejmova za koje je treptanje izvršeno

Na taj način ako broj frejmova poraste na, recimo, 10, dodaćemo 100 poena, na 50 već 1000, a na 100 (što je negde oko 3 sekunde u realnom vremenu) 10.000. Vrednost od 10.000 postavljena je za prag, trenutak kada je potrebno obavestiti vozača o umoru. Sabirak koji se vezuje za frekvenciju treptanja takođe dobijamo pomoću broja frejmova, ali ovog puta broj uzastopnih frejmova u kojima vozač trepće frekvencijom većom nego uobičajeno, i zbira koji se kontinualno računa. Pošto je brojač frejmova počeo da broji od trenutka kada je detektovana prva frekvencija koja je veća od uobičajene (kako je to objašnjeno u prethodnom poglavlju) i inkrementira se sve dok se frekvencija ne vrati “u normalu”, za to vreme se računaju bodovi nakon svake određene frekvencije i dodaju se na zbir koji se pamti. Bodovi zavise od intervala između dva treptaja. Kada se frekvencija odredi, računa se faktor koji dodajemo na zbir i to tako što dvostruku vrednost praga podelimo dobijenom frekvencijom. Ovim je obezbeđeno da što je manja frekvencija na zbir dodajemo više poena. U trenutku kada je detektovano da je frekvencija u okvirima prihvatljivih vrednosti imamo ukupan broj frejmova. Sledeći korak je da taj broj frejmova pomnožimo sa zbirom u koji smo čuvali poene za svaku frekvenciju određenu u međuvremenu.

Time smo u obzir uzeli oba faktora. Sada tu dobijenu vrednost dodajemo na ukupan zbir poena čime se računica završava. Zbir upoređujemo sa postavljenim vrednostima i ukoliko ima potrebe alarmiramo vozača.

Zaključak

Naglo razvijanje automobilske industrije dovelo je do toga da sve više ljudi poseduje automobil. Samim tim je i na ulici sve više vozila, što prouzrokuje i veće gužve, te ljudi za volanom provode mnogo više vremena nego ranije. Međutim, to je dovelo do problema sa prečestim dešavanjem saobraćajnih nesreća. Njihov broj je iz godine u godinu sve veći, zbog čega predstavljaju jedan od glavnih problema sa kojima se današnje društvo susreće. Umoran vozač nije pretnja samo za svoj život nego i za živote drugih učesnika u saobraćaju. Stoga bi bilo korisno pronaći način za otkrivanje umora i pospanosti pre njihovih javljanja i na vreme upozoriti vozača. Teško je dokazati da je do sudara došlo zbog umora vozača. Nije moguće testirati vozače na umor, a vozač koji se sudario možda neće priznati da vozi umoran. Ako policija posumnja na umor, može da ispita dužinu vožnje, način života (da li je vozač spavao dovoljno vremena pre vožnje) i vrstu udarca (mnogi udari zbog umora su velikom brzinom i ne uključuju kočenje jer vozač spava). Nedostatak tragova proklizavanja od kočenja, informacije sa kamera i izjave očevidaca mogu pomoći da se identifikuje umor kao doprinos saobraćajnoj nesreći. Metode za detektovanje umora imaju sve važniju ulogu u sprečavju nesreća i privlače sve više pažnje među istraživačima, ljudi u auto industriji, pa čak i vladinih organizacija.

U ovom radu demonstrirano je kako bi mogao biti implementiran sistem za detekciju umora vozača u realnom vremenu na osnovu obrazaca treptanja, korišćenjem kamere na instrument tabli. Korišćen je prilagođeni matematički model koji određuje nivo umora, a samim tim i nivo alarmantosti situacije, na osnovu učestalosti treptanja i vremena potrebno da se jednom trepne. Lice vozača se analizira tokom svakog frejma, računaju se parametri koji se dalje šalju u algoritam koji računa poene alarmantnosti. Program je razvijen korišćenjem *python* programskog jezika, a za detekciju lica i očiju korišćeni su *OpenCV* i *dlib* biblioteke. Rezultati pokazuju da bi ovakav sistem itakako mogao biti primenjen u praksi, iako postoje razlike u ponašanju prilikom povećava nivoa umora kod osoba.

Ovaj sistem mogao bi da se unapredi, ukoliko se napravi kombinacija sa pomenutim metoda za detektovanje umora, praćenjem fiziološkog stanja vozača i informacija o vozilu. Time bismo dobili verovatno i najprecizniji algoritam, sa najmanjom stopom greške, ali sa druge strane, najskuplji i najzahtevniji po pitanju opreme.

ADAS (Advanced driver-assistance systems) tehnologija, poput sistema za detekciju pospanosti vozača, danas postaje standard u automobilima. U 2015. godini, američki proizvođači automobila su se obavezali da će imati mnogo tipova ADAS opreme kao standard do 2022. Godine. Tehnologija postaje vitalnija za pomoć vozačima i povećanje bezbednosti na putu. Neki veruju da će ova tehnologija takođe postaviti put ka punoj autonomiji, što će u idealnom svetu vožnju učiniti još bezbednijom. Kada budemo imali potpuno autonomne automobile, moći ćemo da odložimo brigu o vozačima koji spavaju u mikrosnu i pospanim vozačima. Međutim, do tada, gubljenje pažnje za volanom može biti smrtonosna greška.

Literatura

1. What are driver drowsiness detection systems and how do they work? - <https://www.tomtom.com/newsroom/explainers-and-insights/driver-drowsiness-detection-systems/>, [poslednji pristup: 10.02.2023.]
2. Automatic Detection of Driver Fatigue Using Driving Operation Information for Transportation Safety - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5492517/>, [poslednji pristup: 13.02.2023.]
3. System and Method for Driver Drowsiness Detection Using Behavioral and Sensor-Based Physiological Measures - <https://www.mdpi.com/1424-8220/23/3/1292>, [poslednji pristup: 11.02.2023.]
4. Real-time embedded eye detection system - <https://www.sciencedirect.com/science/article/pii/S0957417422000070>, [poslednji pristup: 15.02.2023.]
5. Eye blink detection with OpenCV, Python, and dlib - <https://www.geeksforgeeks.org/eye-blink-detection-with-opencv-python-and-dlib/>, [poslednji pristup: 02.02.2023.]
6. Wen-Bing Horng, Chih-Yuan Chen, Yi Chang; Driver Fatigue Detection Based on Eye Tracking and Dynamic' Template Matching, Department of Computer Science and Information Engineering Tamkang University, [poslednji pristup: 18.02.2023.]