

Automatsko raspoznavanje sudačkih odluka u rukometu

Pejatović, Luka

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:432928>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

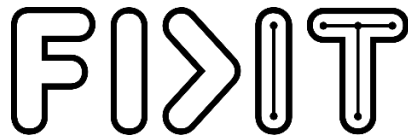
Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)





Sveučilište u Rijeci

**Fakultet informatike
i digitalnih tehnologija**

Sveučilišni diplomski studij Informatika

Luka Pejatović

Automatsko raspoznavanje sudačkih odluka u rukometu

Diplomski rad

Mentor: prof. dr. sc. Marina Ivašić-Kos

Rijeka, kolovoz 2024



Sveučilište u Rijeci

Fakultet informatike
i digitalnih tehnologija

UNIRI



Rijeka, 15.2.2024.

Zadatak za diplomski rad

Pristupnik/ica: Luka Pejatović

Naziv diplomskog rada: **Automatsko raspoznavanje sudačkih odluka u rukometu**

Naziv diplomskog rada na eng. jeziku: **Automatic recognition of referee decisions in handball**

Sadržaj zadatka:

Proučiti i opisati osnovne koncepte dubokih neuronskih mreža s naglaskom na arhitekturu konvolucijske neuronske mreže, aktivacijske funkcije i hiperparametre, te popularne arhitekture konvolucijskih mreža koje se koriste kod rješavanja zadataka računalnog vida kao što je detekcija gesti.

Istražiti biblioteke koje se koriste za učenje modela neuronske mreže i odabrati arhitekturu neuronske mreže koja će se koristiti u specifičnom zadatku detekcije sudačkih gesti u rukometnoj igri.

Prikupiti slike s javno dostupnih skupova slika ili napraviti vlastiti skup slika na kojima će se učiti ili dodatno podesiti model za detekciju sudačkih gesti. Odabrati odgovarajuću arhitekturu i parametre te opisati postupak učenja modela za detekciju sudačkih gesti u rukometu. Koristeći standardne metrike evaluirati rezultate i objasniti najbolji rezultat koji je postignut.

Mentor/ica:
Prof. dr. sc. Marina Ivašić-Kos

Komentor/ica:

Voditeljica za diplomske radove:
Doc. dr. sc. Lucia Načinović Prskalo

Zadatak preuzet: 15.02.2024.

(potpis pristupnika/ce)

Sažetak

Ovaj diplomski rad bavi se temom automatskog raspoznavanja sudačkih odluka u rukometu, s ciljem razvoja sustava koji može prepoznati sudačke gestikulacije i odluke pomoću računalnog vida i umjetne inteligencije. U radu su korišteni različiti algoritmi strojnog učenja kako bi se ostvarila točna klasifikacija različitih sudačkih signala. Za treniranje modela korištene su slike prikupljene iz različitih videozapisa rukometnih utakmica, a potom su modeli testirani na novim slikama kako bi se evaluirala njihova učinkovitost i preciznost. Rezultati istraživanja pokazali su umjeren stupanj točnosti u prepoznavanju sudačkih signala. Dok su neki jednostavniji signali prepoznati s relativno dobrom preciznošću, složenije gestikulacije predstavljale su veći izazov, što ukazuje na potrebu za daljnjim unaprjeđenjem modela kako bi se poboljšala ukupna točnost i preciznost sustava. Zaključak rada ukazuje da je moguće automatizirati proces prepoznavanja sudačkih odluka, ali daljnje istraživanje, kvalitetniji podaci te poboljšanje modela je potrebno kako bi se sustav unaprijedio.

Ključne riječi: automatsko raspoznavanje; računalni vid; strojno učenje; sudačke odluke; rukomet; umjetna inteligencija; klasifikacija signala;

SADRŽAJ

1. Uvod	2
2. Teorijski okvir	3
2.1. Računalni vid i strojno učenje.....	3
2.2. Sudačke odluke u rukometu.....	3
3. Podaci	6
3.1. Priprema podataka	6
4. Tehnologije korištene u prepoznavanju sudačkih gesti	7
4.1. CNN.....	7
4.1.1. Ulazni podaci	7
4.1.2. Konvolucijski slojevi	8
4.1.3. Aktivacijske funkcije	8
4.1.4. Pooling slojevi	8
4.1.5. Potpuno povezani slojevi	8
4.1.6. Izlazni sloj.....	8
4.1.7. Treniranje.....	9
5. Modeli.....	9
5.1. Unaprijed obučeni modeli.....	10
5.1.1. VGG16.....	11
5.1.2. MobileNetV2	12
6. Tehnike analiziranja točnosti modela	14
7. Rezultati modela	15
7.1. Rezultati klasičnog CNN-a	15
7.2. Rezultati modela VGG16.....	17
7.3. Rezultati modela MobileNet	19
8. Optimizacija modela	21
8.1. Optimizacija klasičnog CNN-a	22
8.2. Optimizacija modela VGG16	25
8.3. Poboljšanje modela MobileNet.....	28
9. YOLO	31
9.1. Treniranje modela	32
9.2. Rezultati	32

10. Rasprava.....	35
11. Nadograđivanje sustava	36
12. Zaključak.....	36
Literatura.....	37
Popis tablica	39
Popis slika	40

1. Uvod

Automatsko raspoznavanje sudačkih odluka u rukometu predstavlja inovativnu temu koja objedinjuje različita polja poput računalnog vida i strojnog učenja sa sportskom analizom. Cilj ovog rada je razvit sustav koji koristi algoritme strojnog učenja kako bi automatski prepoznao različite sudačke signale tokom rukometne utakmice. Ova tema je zanimljiva iz razloga što kombinira različite aspekte računalnih znanosti s dinamičnim svijetom sporta. Na taj način može se značajno unaprijediti analiza te samim time povećati transparentnost sportskih događaja za vrijeme utakmice.

Računalni vid i strojno učenje, kao grane umjetne inteligencije, koriste se u raznim područjima poput medicinske dijagnostike, razvijanje autonomnih vozila i sigurnosnih sustava. Njihova primjena u sportu, za sada, još uvijek je nedovoljno istražena te primijenjena. U kontekstu rukometa, koji se nalazi među zahtjevnijim sportovima, što zbog brzine, što zbog fizičke snage, prepoznavanje sudačkih odluka predstavlja veliki izazov zbog složenih i brzih gestikulacija sudaca. Razvijanje sustava koji se bavi prepoznavanjem sudačkih odluka jedan je načina na koji se može pridonijeti sportskoj analitici. Pomoću ovog sustava ubrzao bi se i unaprijedio način na koji se bilježe sudačke odluke, koje kasnije služe za analizu. Također, u skladu sa sustavom koji može sam prepoznati ispravnu odluku, smanjila bi se subjektivnost u evaluaciji sudačkih odluka. Isto tako, dokumentacija svih sudačkih znakova može biti korisna za obuku novih sudaca. Oni mogu pregledavati automatski prepoznate geste i učiti kako ih pravilno koristiti interpretirati, čime se doprinosi njihovom profesionalnom razvoju i razumijevanju pravila.

Motivacija za odabir ove teme dolazi iz interesa za tehnologije računalnog vida i strojnog učenja te njihovu primjenu u sportu, s posebnim naglaskom na sportsku analitiku. Iskustvo u rukometu, stečeno kroz različite uloge – igrača, sudca, zapisničara i trenera – omogućilo je uvid u mnoge aspekte ovog sporta. Temeljem tog iskustva, težnja je pridonijeti razvoju rješenja koja bi mogla olakšati praćenje utakmica, poboljšati sudačku učinkovitost te povećati transparentnost u sportu. Cilj je pokazati kako tehnologija može igrati ključnu ulogu u modernizaciji i profesionalizaciji sporta.

2. Teorijski okvir

2.1. Računalni vid i strojno učenje

Za bolje razumijevanje ovoga rada potrebno je pojasniti ključne pojmove poput računalnog vida i strojnog učenja. Za početak, računalni vid je polje umjetne inteligencije (AI) koje koristi strojno učenje i neuronske mreže kako bi naučilo računala i sustave da izvuku značajne informacije iz digitalnih slika, videa i drugih vizualnih inputa i da interpretiraju informaciju sadržanu u slici i daju preporuke ili poduzmu radnje kada vide nedostatke ili probleme [1]. Dakle računalni vid funkcionira na isti način kao i ljudski, samo što ljudi imaju tu prednost iskustva. Ljudsko oko ima cijeli ljudski život kako bi učilo i naučilo raspoznati određene stvari, kako izgledaju i koliko su udaljenije, dok računalo to radi u znatno kraćem vremenskom razdoblju. Kako bi to sve uspješno funkcioniralo, potrebno je jako puno podataka. Na njima se provodi učenje sve dok ne postigne odgovarajuće performanse ili dok se ne iscrpe svi podaci koje model može naučiti iz tih podataka.

Strojno učenje (ML) grana je umjetne inteligencije (AI) i računalne znanosti koja se usredotočuje na korištenje podataka i algoritama kako bi omogućila AI da oponaša način na koji ljudi uče, postupno poboljšavajući svoju točnost [1]. Razlikujemo tri vrste strojnog učenja: Nadzirano učenje (engl. *supervised learning*), nenadzirano učenje (engl. *unsupervised learning*) i učenje s povratnom vezom (engl. *reinforced learning*). U ovom radu koristit ćemo nadzirano učenje.

U nadziranom učenju, model se trenira na označenim podacima, gdje su ulazi povezani s odgovarajućim izlazima (etiketama). Cilj je naučiti funkciju koja povezuje ulaz s izlazom, kako bi se na temelju novih ulaza moglo predvidjeti ispravne izlaze. Neki od primjera upotrebe nadziranog učenja su klasifikacije slika, prepoznavanje govora ili predviđanje cijena.

Nenadzirano učenje, za razliku od nadziranog učenja, ne koristi unaprijed označene podatke. Na modelu je da on sam prepozna skrivene strukture ili otkrije odnose između podataka. Dakle izlaz takvog modela su podaci koji su grupirani na način da međusobno imaju nešto zajedničko. Primjeri su klasteriranje podataka, otkrivanje anomalija i smanjenje dimenzionalnosti.

Učenje s povratnom vezom temelji se na konceptu nagrađivanja i kažnjavanja. Model „agent“ uči optimalne radnje u nekom okruženju kroz interakciju s njim. Radnje koje pridonose cilju modela su pojačane dok radnje koje ga odvrćaju se zanemaruju.

2.2. Sudačke odluke u rukometu

Suđenje u rukometu je jedan od ključnih aspekta igre. Bez ispravnog navođenja rukometnih pravila, natjecanje ne bi imalo pravilni oblik natjecanja. Suci tokom igre koriste niz različitih gesti kako bi označili odgovarajuće radnje. Zbog brze igre, suci u rukometu su primorani u jako kratkom vremenskom roku donijeti ispravnu odluku. Uz to sama igra ima puno fizičkog kontakta što dodatno otežava donošenje odluke. Osim tehničke preciznosti, suci moraju

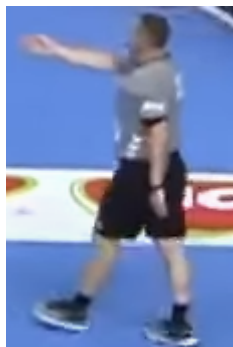
održavati autoritet na terenu i osigurati disciplinu među igračima i trenerima. Uvođenjem tehnologija poput VAR (Video assistant referee), predstavlja korak prema unaprjeđenju igre te smanjuje subjektivnost sudačkih odluka.

Sve ukupno postoji 18 službenih sudačkih gesti, dok uz njih još postoje različite dopunske geste koje detaljnije opisuju situaciju. Za potrebu ovog rada korištene su sljedeće geste :

- Gol (slika 1)
- Sedmerac (slika 2)
- Time-out (slika 3)
- Pasivni napad (slika 4)
- Isključenje (slika 5)



Slika 1. Gol



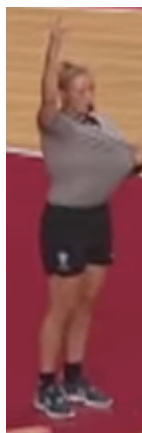
Slika 2. Sedmerac



Slika 3. Time-out



Slika 4. Pasivni napad



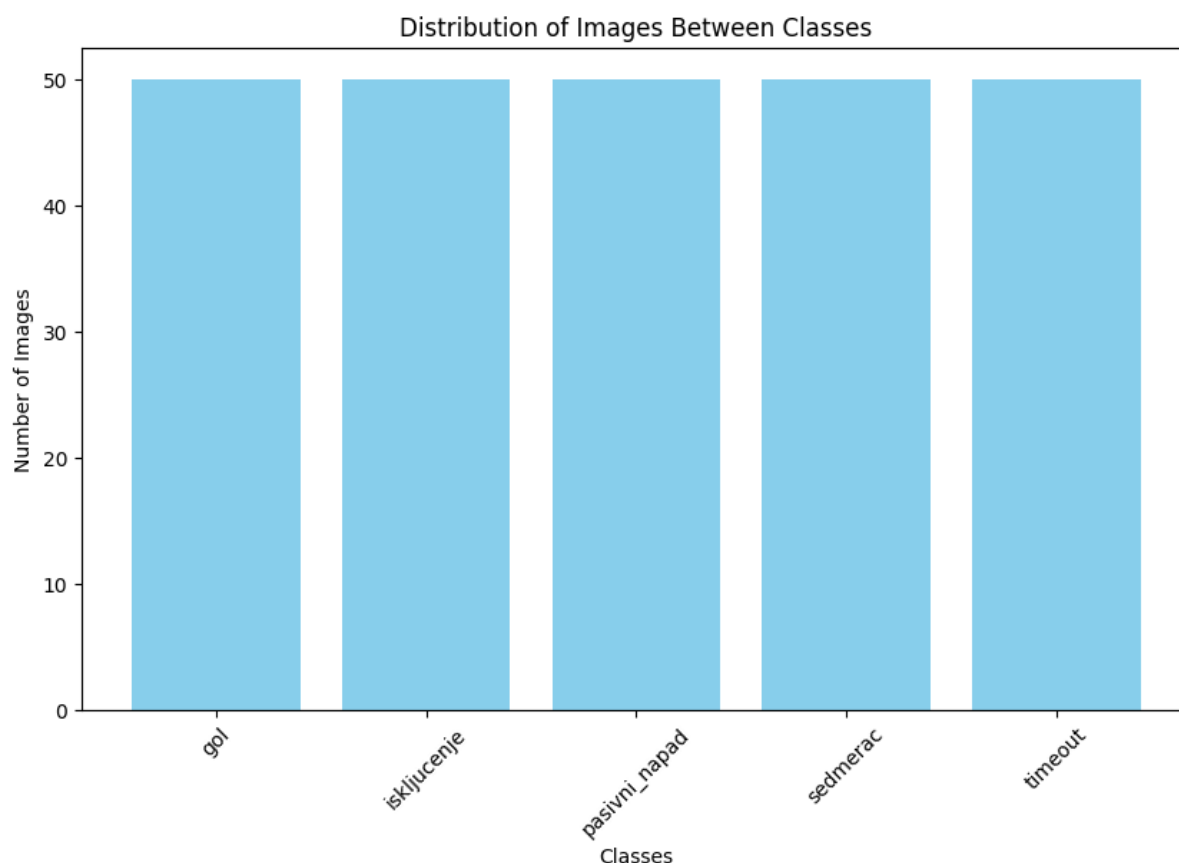
Slika 5. Isključenje

3. Podaci

Prikupljane podataka jedan je od najvažnijih koraka za izradu kvalitetnog modela. Svi podaci, odnosno slike ručno su odabrane, kroz izrezivanje slike iz različitih videozapisa rukometnih utakmica. Dakle, svaka pojedinačna slika izrezana je na način da prikazuje samo tu određenu sudačku gestu. Kako bi učinili slike raznovrsnije uzete su iz različitih kutova, time osiguravamo da se model uči nositi sa različitim scenarijima u stvarnom vremenu. Nakon izrezivanja , slike su potom spremljene u odgovarajuću mapu sa odgovarajućom sudačkom odlukom. Kao što je spomenuto ranije, kada koristimo nadzirano učenje jako je bitno da sve slike budu svrstane u mapu sa pravim nazivom.

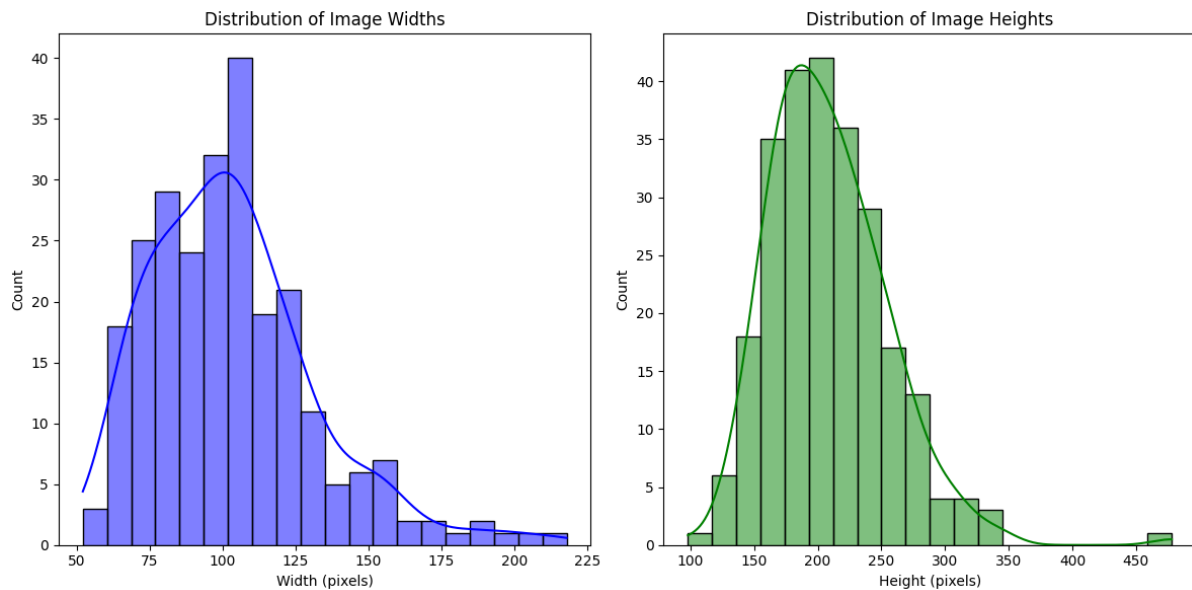
3.1. Priprema podataka

Skup podatka sastoji se od sveukupno 250 slika jednako raspoređene na svih 5 sudačkih odluka odnosno klasa, što je vidljivo na slici 6. Na taj način osigurali smo da nam cjelokupni skup bude uravnotežen te da model preciznije klasificira svaku klasu.



Slika 6. Raspodjela klasa

Nakon što smo prikupili sve podatke razvrstali ih mape, potrebno je provjeriti njihove oblike, odnosno dimenzije. Na slici 7. vidimo distribuciju pixela svih slika kroz njihovu širinu i visinu. Možemo uočiti da većina slika kreće oko 100 piksela u širinu, te najviše ih ima sa širinom od 110. Na desnoj strani slike vidimo visinu slika, te se velika većina kreće oko 200 piksela. Ovi podaci su nam jako bitni za izgradnju modela, pošto za model je potrebno definirati jedinstvenu veličinu za sve slike.



Slika 7. Dimenzije slika

4. Tehnologije korištene u prepoznavanju sudačkih gesti

4.1. CNN

Računalni vid obuhvaća različite tehnike koje omogućuju računalima analizu i interpretaciju vizualnih podataka iz stvarnog svijeta. Jedna od glavnih tehnika za prepoznavanje vizualnih obrazaca u slikama i videozapisima je CNN (Convolutional Neural Networks), odnosno konvolucijske neuralne mreže. Razlog zbog čega je ova tehnika uspješna je upravo taj što imaju sposobnost da automatski uče značajke iz podataka kroz slojeve konvolucija i filtracije.

4.1.1. Ulazni podaci

CNN uzima slike kao podatke. Svaku sliku možemo prikazati kao niz piksela, gdje svaki piksel ima vrijednost koja predstavlja svjetlinu ili boju (ako se radi o RGB slikama, svaki piksel ima tri kanala: crveni, zeleni i plavi). Što znači da sve slike u prikupljenom skupu podataka imaju oblik npr. 100x200x3.

4.1.2. Konvolucijski slojevi

Konvolucijski sloj je osnovni dio CNN-a i obavlja posao konvolucije. Cilj ovoga sloja je izdvajanje značajki iz slike pomoću filtera. To je mala matrica koja najčešće ima oblik 3x3 ili 5x5 koja se pomiče preko slike i multiplicira i zbraja piksele ispod filtera. Filter izdvaja važne uzorke, poput rubova ili oblika. Dok filter se pomiče po cijeloj slici, primjenjujući se na svaki dio slike, stvara mapu značajki (engl. *feature map*). U početnim slojevima, filter prepoznaje jednostavne značajke, dok u dubljim slojevima prepoznaje složenije obrasce, poput dijelova objekata ili čak cijele objekte.

4.1.3. Aktivacijske funkcije

Nakon svake konvolucije, prolazimo kroz aktivacijsku funkciju. Najčešće se koristi ReLU (Rectified Linear Unit), koja zamjenjuje sve negativne vrijednosti u mapi značajki sa nulom. Na taj način stvara se nelinearnost u mreži, što je ključno za učenje složenih obrazaca.

4.1.4. Pooling slojevi

Pooling slojevi služe za smanjenje dimenzionalnosti mape značajki, što smanjuje broj parametara i računalne složenosti te omogućuje bolju generalizaciju. Najčešće korištena tehnika je maksimalni pooling (Max Pooling). Unutar svake lokalne regije (npr. 2x2 piksela) odabire se maksimalna vrijednost, čime se smanjuje veličina izlaza, a najvažnije informacije zadržavaju. Pooling ne smanjuje samo veličinu slike nego također čini mrežu otpornijom na male promjene i pomake u slici.

4.1.5. Potpuno povezani slojevi

Nakon što se izvuče značajke iz slike kroz niz konvolucijskih i pooling slojeva, izlazna mapa značajki se "poravnava" u jednovektorsku formu, koja se zatim prosljeđuje kroz potpuno povezane slojeve. U potpuno povezanim slojevima, svaki neuron je povezan sa svim neuronima u prethodnom sloju, kao kod klasične neuronske mreže. Ovi slojevi služe za kombiniranje svih naučenih značajki i donošenje konačne odluke

4.1.6. Izlazni sloj

Na kraju mreže obično se koristi softmax sloj za klasifikacijske zadatke. Ovaj sloj pretvara izračunate vrijednosti u vjerojatnosti, dajući konačnu klasifikaciju slike. Softmax funkcija uzima skup vrijednosti i pretvara ih u vjerojatnosti, pri čemu zbroj svih vjerojatnosti iznosi 1. Najveća vjerojatnost daje konačnu klasu slike.

4.1.7 Treniranje

CNN se trenira pomoću algoritma unaprijedenog širenja greške i optimizacije. Tijekom treniranja mreža prolazi kroz veliki skup označenih slika. Mreža uspoređuje svoje predikcije s točnim oznakama i izračunava gubitak, koji mjeri koliko je pogrešna predikcija. Algoritmi optimizacije (poput Adam ili SGD - Stochastic Gradient Descent) koriste ovaj gubitak kako bi prilagodili težine unutar mreže, pokušavajući smanjiti ukupnu pogrešku. Treniranje se ponavlja kroz više epoha, s ciljem da mreža poboljša svoje predikcije i nauči složene obrasce u podacima.

5. Modeli

Za izradu sustava prepoznavanja rukometnih sudačkih gesti korišteni su razni modeli. Za početak imamo klasični CNN model koji sami definiramo sve slojeve i ostale parametre.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.callbacks import ReduceLROnPlateau

def build_cnn_model(input_shape, num_classes):
    model = Sequential([
        Conv2D(32, (3, 3), activation='relu', input_shape=input_shape),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(64, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(128, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Conv2D(256, (3, 3), activation='relu'),
        MaxPooling2D(pool_size=(2, 2)),

        Flatten(),

        Dense(128, activation='relu'),
        Dropout(0.5),

        Dense(num_classes, activation='softmax')
    ])

    return model

model = build_cnn_model(input_shape=(img_height, img_width, 3),
num_classes=len(train_generator.class_indices))

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6)
```

```
history = model.fit(
    train_generator,
    epochs=15,
    validation_data=val_generator,
    callbacks=[reduce_lr]
)
```

Ovaj kod implementira konvolucijsku neuronsku mrežu (CNN) koristeći Keras biblioteku unutar TensorFlow okvira, s ciljem klasifikacije slika. CNN se gradi koristeći Sequential model, koji omogućuje dodavanje slojeva u linearnoj sekvenci. Prvi sloj je Conv2D, koji služi za konvoluciju ulaznih slika (tj. primjenu filtarskih jezgri za izdvajanje značajki poput rubova, tekstura i oblika). Model započinje s 32 filtra veličine 3x3 i koristi ReLU aktivacijsku funkciju. Slijedi MaxPooling2D sloj koji smanjuje dimenzionalnost slike (npr. 2x2 pooling) zadržavajući najvažnije značajke, čime se smanjuje računaska složenost i smanjuje mogućnost pre-treniranja modela.

Nakon toga, dodaju se još tri konvolucijska sloja s povećanjem broja filtara (64, 128, 256), a svaki sloj ponovno prati MaxPooling kako bi se dalje reducirala dimenzionalnost. Nakon ovih slojeva, ulazna slika se pretvara u jednodimenzionalni vektor pomoću „Flatten“ sloja. Slijedi „Dense“ sloj s 128 neurona koji koristi „ReLU“ aktivacijsku funkciju za dodavanje nelinearnosti modelu, te „Dropout“, sloj koji isključuje 50% neurona tijekom treniranja kako bi se smanjilo pre-treniranje.

Konačni sloj je „Dense“ sloj s brojem neurona koji odgovara broju klasa, a koristi softmax aktivaciju za klasifikaciju. Model se trenira korištenjem Adam optimizatora i „categorical_crossentropy“ funkcije gubitka, koja je standardna za više-klasne klasifikacijske zadatke. Tokom treniranja koristi se „ReduceLROnPlateau callback“ funkcija, koja smanjuje stopu učenja ako se gubitak na validacijskom skupu ne poboljša nakon tri epohe, što pomaže u boljoj optimizaciji modela. Model se trenira na skupu podataka pomoću „train_generator“ i validira na „val_generator“, pri čemu se broj epoha postavlja na 15.

5.1. Unaprijed obučeni modeli

Unaprijed obučeni modeli su modeli strojnog učenja koji su već trenirani na velikim i raznovrsnim skupovima podataka, omogućujući bržu i učinkovitiju primjenu u različitim zadacima. Ovi modeli, poput ResNet-a, VGG-a i MobileNet-a za prepoznavanje slika ili BERT-a za obradu prirodnog jezika, već su razvili sposobnosti prepoznavanja obrazaca i značajki koje se mogu prilagoditi specifičnim potrebama korisnika. Prednosti korištenja unaprijed obučeni modela uključuju brži razvoj aplikacija, manju potrebu za velikim skupovima podataka i često bolje performanse zbog njihovog prethodnog obučavanja na raznovrsnim podacima. Oni se obično koriste za fino podešavanje na specifične zadatke, ekstrakciju značajki ili transfer učenje, gdje se značajke naučene iz jednog zadatka koriste za poboljšanje učenja na drugom. Popularne platforme i biblioteke poput TensorFlow Hub,

PyTorch Hub i Hugging Face Model Hub nude unaprijed obučene modele koji mogu značajno ubrzati razvoj i poboljšati rezultate aplikacija.

5.1.1. VGG16

```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score,
f1_score, recall_score

from tensorflow.keras.preprocessing.image import ImageDataGenerator

main_folder_path = 'znakovi'
image_size = (224, 224)
batch_size = 32
epochs = 15
train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    main_folder_path,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)

validation_generator = train_datagen.flow_from_directory(
    main_folder_path,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)

base_model = VGG16(weights='imagenet', include_top=False, input_shape=(image_size[0],
image_size[1], 3))

x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(train_generator.num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit(
    train_generator,
    validation_data=validation_generator,
    epochs=epochs
)
```



```

model.save('handball_referee_model.h5')

from tensorflow.keras.preprocessing import image
import numpy as np

def predict_referee_signal(img_path, model):
    img = image.load_img(img_path, target_size=image_size)
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array /= 255.0
    prediction = model.predict(img_array)
    predicted_class = np.argmax(prediction[0])
    class_labels = train_generator.class_indices
    class_labels = {v: k for k, v in class_labels.items()}

    return class_labels[predicted_class]

validation_generator.reset()
predictions = model.predict(validation_generator)

```

Ovdje vidimo kod koji koristi TensorFlow i Keras za obuku konvolucijskog neuronskog mrežnog modela na temelju unaprijed obučene arhitekture VGG16. Prvo se definiraju putanje i parametri za obuku, uključujući veličinu slika, veličinu serije (engl. batch size) i broj epoha. Potom se koristi „ImageDataGenerator“ za augmentaciju i pred procesiranje slika, što uključuje normalizaciju i različite transformacije poput rotacije i horizontalnog okretanja. Ovaj generator također razdvaja podatke na obučavajuće i validacijske skupove. Nakon toga, učitava se unaprijed obučeni model VGG16 bez gornjih slojeva, a na njega se dodaju prilagođeni slojevi za klasifikaciju, uključujući globalno prosječno zgrušavanje značajki i potpuno povezane slojeve. Uslijedilo je zamrzavanje težina prethodno obučениh slojeva kako bi se spriječilo njihovo treniranje tijekom početne obuke, što omogućava modelu da se fokusira na prilagodbu novih slojeva. Model se zatim kompajlira koristeći Adam optimizator i kategorijsku unakrsnu entropiju kao funkciju gubitka, a obuka se provodi kroz zadani broj epoha. Nakon obuke, model se sprema za kasniju upotrebu, dok se za predikcije koristi funkcija koja učitava slike, normalizira ih i predviđa klase koristeći obučeni model.

5.1.2 MobileNetV2

```

data_dir = 'znakovi'

img_height, img_width = 224, 224
batch_size = 32

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

test_datagen = ImageDataGenerator(rescale=1./255)

```

```

temp_generator = train_datagen.flow_from_directory(
    data_dir,
    target_size=(img_height, img_width),
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

file_paths = temp_generator.filepaths
labels = temp_generator.classes
class_indices = temp_generator.class_indices

class_names = {v: k for k, v in class_indices.items()}
string_labels = [class_names[label] for label in labels]

train_paths, temp_paths, train_labels, temp_labels = train_test_split(
    file_paths, string_labels, test_size=0.4, stratify=string_labels, random_state=seed
)

val_paths, test_paths, val_labels, test_labels = train_test_split(
    temp_paths, temp_labels, test_size=0.5, stratify=temp_labels, random_state=seed
)

def create_generator(file_paths, labels, datagen, batch_size):
    df = pd.DataFrame({'filename': file_paths, 'class': labels})
    generator = datagen.flow_from_dataframe(
        dataframe=df,
        x_col='filename',
        y_col='class',
        target_size=(img_height, img_width),
        batch_size=batch_size,
        class_mode='categorical',
        shuffle=True
    )
    return generator

train_generator = create_generator(train_paths, train_labels, train_datagen, batch_size)
val_generator = create_generator(val_paths, val_labels, test_datagen, batch_size)
test_generator = create_generator(test_paths, test_labels, test_datagen, batch_size)

base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(img_height,
img_width, 3))
base_model.trainable = False
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x)
x = Dropout(0.5)(x)
output = Dense(5, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-6)

history = model.fit(
    train_generator,
    epochs=10,
    validation_data=val_generator,
    callbacks=[reduce_lr]
)

```

Prvo se definira putanja do skupa podataka i parametri za obrada slika, uključujući visinu, širinu i veličinu serije. Zatim se postavlja augmentacija podataka za obučavanje koristeći „ImageDataGenerator“, koja uključuje transformacije poput rotacije i horizontalnog okretanja, dok za validaciju i testiranje koriste samo normalizaciju. Učitava se cijeli skup podataka da bi se dobili indeksi klasa i putanje do slika, nakon čega se klase mapiraju na njihove „string“ etikete. Skup podataka se zatim dijeli na obučavajući, validacijski i testni dio koristeći „train_test_split“ iz „scikit-learn-a“, pri čemu ravnoteža klasa kroz stratifikaciju ostaje očuvana. Funkcija „create_generator“ generira podatke iz skupa podataka s putanjama i etiketama koristeći „flow_from_dataframe“, što omogućava dinamičko učitavanje slika i pripadajućih oznaka. Model se gradi koristeći MobileNetV2, unaprijed obučeni model koji se koristi bez gornjih slojeva. Dodaju se slojevi za globalno prosječno zgrušavanje, potpuno povezani sloj s ReLU aktivacijom, dropout sloj za regularizaciju, i izlazni sloj s softmax aktivacijom za klasifikaciju u 5 klasa. Model se kompajlira s Adam optimizatorom i kategorijskom unakrsnom entropijom kao funkcijom gubitka. Kako bi se poboljšala obuka, koristi se „ReduceLROnPlateau callback“ za smanjenje stope učenja kada se validacijski gubitak ne poboljšava. Model se zatim obučava kroz 10 epoha koristeći obučavajući i validacijski generator.

6. Tehnike analiziranja točnosti modela

Analiziranje točnosti modela je ključni korak u ocjeni njegovih performansi i učinkovitosti. Postoji nekoliko alata i tehnika koji omogućuju dubinsku analizu kako bi se razumjela preciznost i pouzdanost modela. Matricu konfuzije koristi se za prikaz stvarnih nasuprot predviđenim klasama, omogućujući vizualizaciju točnih i pogrešnih predikcija za svaku klasu. Na temelju ove matrice, mogu se izračunati ključne metrike kao što su preciznost (udio ispravno predviđenih pozitivnih slučajeva u odnosu na sve pozitivne predikcije), „recall“ (udio ispravno predviđenih pozitivnih slučajeva u odnosu na sve stvarne pozitivne slučajeve) i „F1-score“ (sredina preciznosti i „recalla“ koja daje uravnoteženu ocjenu modela).

Uz to, ROC (Receiver Operating Characteristic) krivulja i AUC (Area Under the Curve) pomažu u evaluaciji sposobnosti modela da razlikuje između pozitivnih i negativnih klasa pri različitim pragovima, pružajući sveobuhvatan uvid u njegovu generalizaciju. Unakrsna validacija koristi se za procjenu robusnosti modela kroz višestruko dijeljenje podataka na trening i test skupove, što pomaže u identificiranju moguće pre treniranja i procjeni performansi na različitim podskupovima podataka. Također, grafički prikazi poput krivulje učenja omogućuju praćenje promjena u točnosti i gubitku tijekom treniranja, što može pomoći u prepoznavanju problema kao što su pretreniranje ili nedovoljno treniranje. Korištenjem ovih alata, istraživači i inženjeri mogu detaljno analizirati performanse modela, identificirati njegove slabosti i usmjeriti daljnje korake za poboljšanje preciznosti i pouzdanosti.

7. Rezultati modela

Rezultati treniranja konvolucijske neuronske mreže (CNN) u ovom radu pokazuju kako se model ponašao u prepoznavanju sudačkih gesti u rukometu. Nakon treniranja modela na skupu podataka koji se sastojao od različitih sudačkih signala, ostvareni rezultati ukazuju na točnost prepoznavanja od oko 50 do 60 posto. Ova razina točnosti ukazuje na to da model u prosjeku uspješno prepoznaje polovinu do dvije trećine sudačkih gesti ispravno. Ova razina točnosti ukazuje na to da model u prosjeku uspješno prepoznaje polovinu sudačkih gesti ispravno. Ovi rezultati su postignuti primjenom standardnih konvolucijskih slojeva, slojeva za smanjenje dimenzionalnosti i potpuno povezanih slojeva.

7.1. Rezultati klasičnog CNN-a

Kada gledamo metrike za evaluaciju na slici 8., možemo vidjeti učinkovitost u prepoznavanju svih 5 sudačkih gesti. Preciznost testa iznosi 38% , što ukazuje da model od svih instanci u testnom skupu ispravno klasificira 38%. Kada gledamo pojedinačno za svaku klasu rezultati su sljedeći:

- **Gol:** Preciznost iznosi 0.27, što znači da je 27% svih predviđenih "golova" ispravno identificirano. „recall“ je 0.40, što ukazuje da je 40% svih stvarnih golova ispravno prepoznato. Na kraju imamo uravnoteženi prikaz u obliku F1-score-a koji iznosi 32%.
- **Isključenje:** S preciznošću od 0.17 i „recallom“ od 0.10, model pokazuje nisku sposobnost u prepoznavanju ove geste, s F1-score-om od 0.12, što sugerira da je učinkovitost niska.
- **Pasivni napad:** Preciznost je 0.33, dok je „recall“ 0.10, što označava da model često promašuje ovu gestu, s F1-score-om od 0.15.
- **Sedmerac:** Ova klasa ima preciznost od 0.12 i „recall“ od 0.20, s F1-score-om od 0.15, što pokazuje da model ima značajne poteškoće u prepoznavanju ove geste.
- **Timeout:** Preciznost i „recall“ za ovu klasu su oba 0.20, s F1-score-om od 0.20, ukazujući na sličan nivo učinkovitosti kao kod drugih klasa.

„Macro avg“ i „weighted avg“ metričke vrijednosti su 0.22 za preciznost, 0.20 za „recall“ i 0.19 za F1-score, što pokazuje prosječnu izvedbu modela na svim klasama. Ove prosječne vrijednosti sugeriraju da model ima problema s generalizacijom i prepoznavanjem svih klasa s jednakim učinkom

```

Test accuracy: 0.3799999952316284
2/2 ————— 0s 146ms/step
      precision    recall  f1-score   support

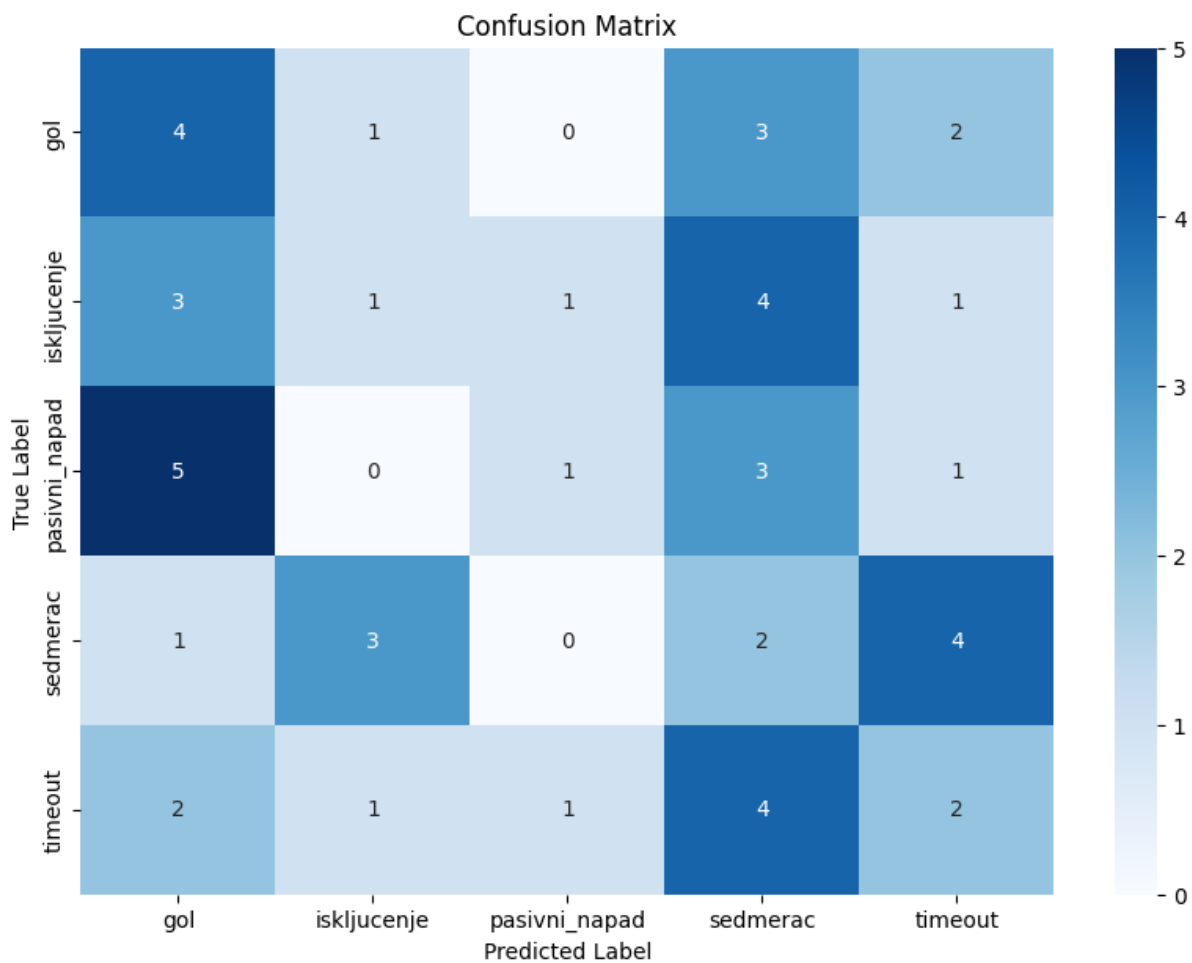
   gol           0.27     0.40     0.32         10
  iskljucenje    0.17     0.10     0.12         10
 pasivni_napad   0.33     0.10     0.15         10
   sedmerac     0.12     0.20     0.15         10
   timeout      0.20     0.20     0.20         10

 accuracy              0.20         50
  macro avg           0.22     0.20     0.19         50
  weighted avg           0.22     0.20     0.19         50

```

Slika 8. Metrike evaluacije klasičnog CNN-a

Ako preciznije analiziramo rezultate ovoga modela, gledajući matricu konfuzije možemo vidjeti koje klase model predviđa točno a koje klase zamjenjuje sa drugim(slika 9.).



Slika 9. Matrica konfuzije za klasični CNN

Kad gledamo matricu konfuzije, bitno nam je gledati dijagonalu na kojoj se nalaze sve točno predviđene vrijednosti. Na ovom primjeru vidimo da su rezultati na dijagonali poprilično niski što ukazuje da model se muči sa predviđanjem točnih klasa. Najveću vrijednost ima klasa „gol“, što je i očekivano pošto je jedna od najjednostavnijih. Istako tako vidimo da najviše zamjenjuje klasu „gol“ sa klasom „pasivni napad“, što je opet očekivano jer su te dvije klase poprilično slične.

7.2. Rezultati modela VGG16

Na slici 10 možemo vidjeti metrike evaluacije modela VGG16. Preciznost testa iznosi 16%, što znači da je model ispravno klasificirao 16% svih instanci u testnom skupu. Kada detaljnije pogledamo rezultate za svaku klasu, uočavamo sljedeće:

- **Gol:** Preciznost iznosi 0.14, što znači da je 14% svih predviđenih "golova" ispravno identificirano. „recall“ iznosi 0.10, što ukazuje na to da je samo 10% svih stvarnih golova prepoznato ispravno. F1-score za ovu klasu iznosi 0.12, što pokazuje nisku učinkovitost modela u prepoznavanju golova.
- **Isključenje:** Model postiže preciznost od 0.25 i „recall“ od 0.10, što sugerira da se isključenje rijetko prepoznaje točno, s F1-scoreom od 0.14, što ukazuje na još jedan nisko prepoznatljiv signal.
- **Pasivni napad:** Preciznost je 0.11, dok je „recall“ 0.10, što znači da model često promašuje ovu gestu. F1-score je 0.11, što ukazuje na poteškoće modela u prepoznavanju pasivnog napada.
- **Sedmerac:** Ova klasa ima nešto bolji „recall“ od 0.40, što znači da model često prepoznaje sedmerce, ali s preciznošću od 0.15 i F1-scoreom od 0.22, učinkovitost ostaje niska zbog velikog broja lažno pozitivnih predviđanja.
- **Timeout:** Ova klasa ima preciznost od 0.25 i „recall“ od 0.10, s F1-scoreom od 0.14, što sugerira da model također ima poteškoća u ispravnom prepoznavanju timeout gesti.

Prosječne metričke vrijednosti kroz sve klase (macro avg) su 0.18 za preciznost, 0.16 za „recall“ i 0.15 za F1-score, dok „weighted avg“ metričke vrijednosti također odražavaju slične performanse modela, s preciznošću od 0.18, „recallom“ od 0.16 i F1-scoreom od 0.15. Ovi rezultati ukazuju na to da model ima problema s generalizacijom i nisku učinkovitost u prepoznavanju svih klasa sudačkih gesti. Potrebna su dodatna poboljšanja kako bi se postigli bolji rezultati i veća točnost modela.

```

Accuracy: 0.16
Recall: 0.16
F1 Score: 0.14616934493095485

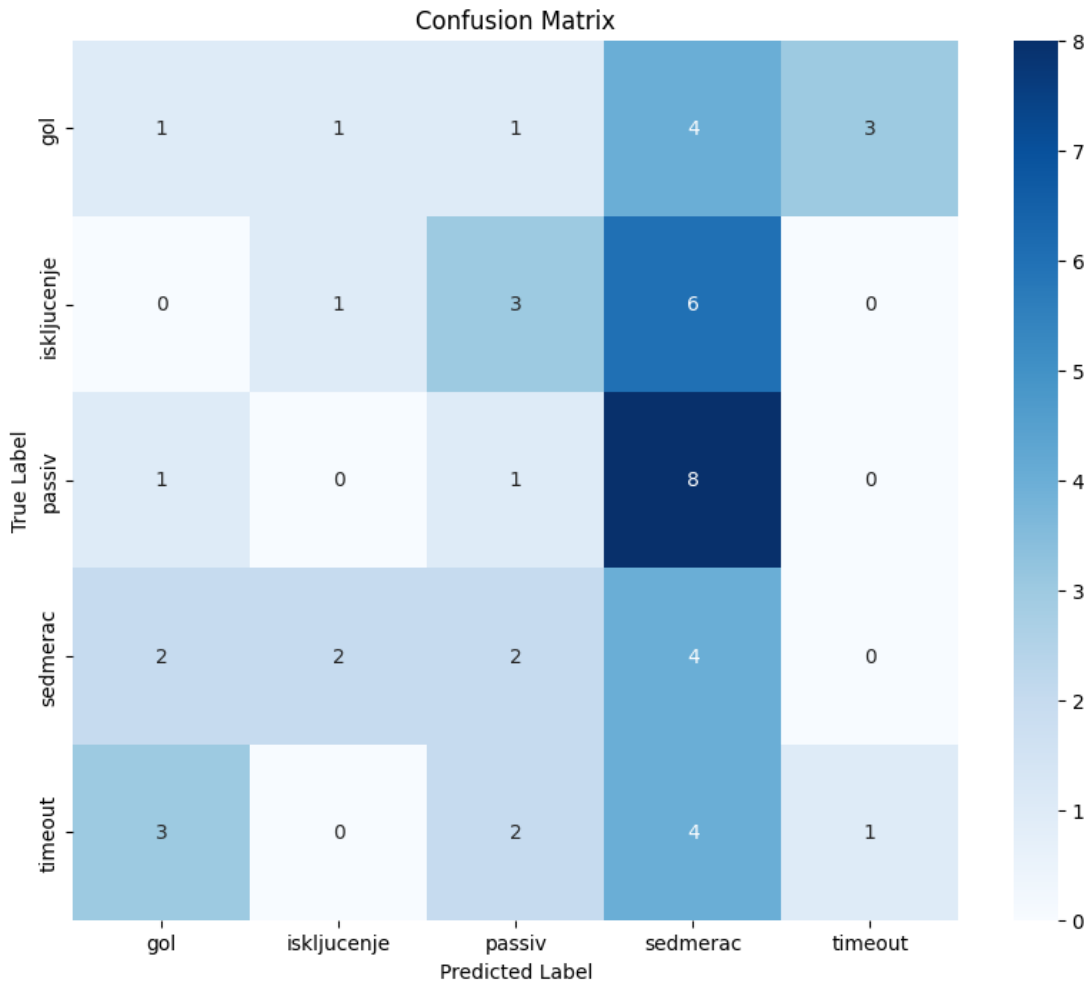
Classification Report:

```

	precision	recall	f1-score	support
gol	0.14	0.10	0.12	10
isključenje	0.25	0.10	0.14	10
passiv	0.11	0.10	0.11	10
sedmerac	0.15	0.40	0.22	10
timeout	0.25	0.10	0.14	10
accuracy			0.16	50
macro avg	0.18	0.16	0.15	50
weighted avg	0.18	0.16	0.15	50

Slika 10. Metrike evaluacije VGG16 modela

Kroz matricu konfuzije (slika 11.) uočavamo da model jedino klasu „sedmerac“ predviđa više od jednom točno. Međutim, isto tako klasu „sedmerac“ je najčešće predviđa kao neku drugu klasu, a najviše „pasivni napad“.



Slika 11. Matrica konfuzije modela VGG16

7.3. Rezultati modela MobileNet

Rezultati evaluacije modela (slika 12.) pokazuju ukupnu točnost od 36%, što znači da model ispravno klasificira 36% instanci iz testnog skupa. Međutim, kada se promatraju pojedinačne klase, rezultati su prilično neujednačeni.

- **Gol:** Model je postigao preciznost od 0.09 i „recall“ od 0.10, što znači da je samo 9% predviđanja za "gol" točno, dok je prepoznato 10% stvarnih slučajeva golova. F1-score iznosi 0.10, što ukazuje na nisku učinkovitost u prepoznavanju ove geste.
- **Isključenje:** Ova klasa pokazuje najnižu učinkovitost, s preciznošću, „recallom“ i F1-scoreom od 0.00, što znači da model nije uspio prepoznati niti jedno isključenje iz testnog skupa.
- **Pasivni napad:** Preciznost je 0.18, a „recall“ 0.20, što znači da model ponekad ispravno prepoznaje pasivne napade, ali s F1-scoreom od 0.19, performanse ostaju niske.
- **Sedmerac:** Ova klasa ima relativno bolji „recall“ od 0.50, što znači da je model uspio prepoznati polovicu svih sedmeraca. Međutim, preciznost od 0.24 i F1-score od 0.32 sugeriraju da je točnost prepoznavanja još uvijek sub optimalna.

- **Timeout:** Za ovu klasu model nije postigao nikakve uspješne predikcije, s preciznošću, „recallom“ i F1-scoreom od 0.00.

„Macro avg“ i „weighted avg“ vrijednosti dodatno potvrđuju ove rezultate, s prosječnim preciznostima od 0.10, „recallom“ od 0.16 i F1-scoreom od 0.12. Ovi rezultati pokazuju da je model vrlo neefikasan u prepoznavanju većine gesti i da su potrebna značajna poboljšanja, poput proširenja i poboljšanja skupa podataka, optimizacije hiperparametara, te korištenja naprednijih tehnika treniranja kako bi se postigla bolja točnost i učinkovitost.

```

Test accuracy: 0.36000001430511475
2/2 ----- 2s 1s/step
      precision    recall  f1-score   support

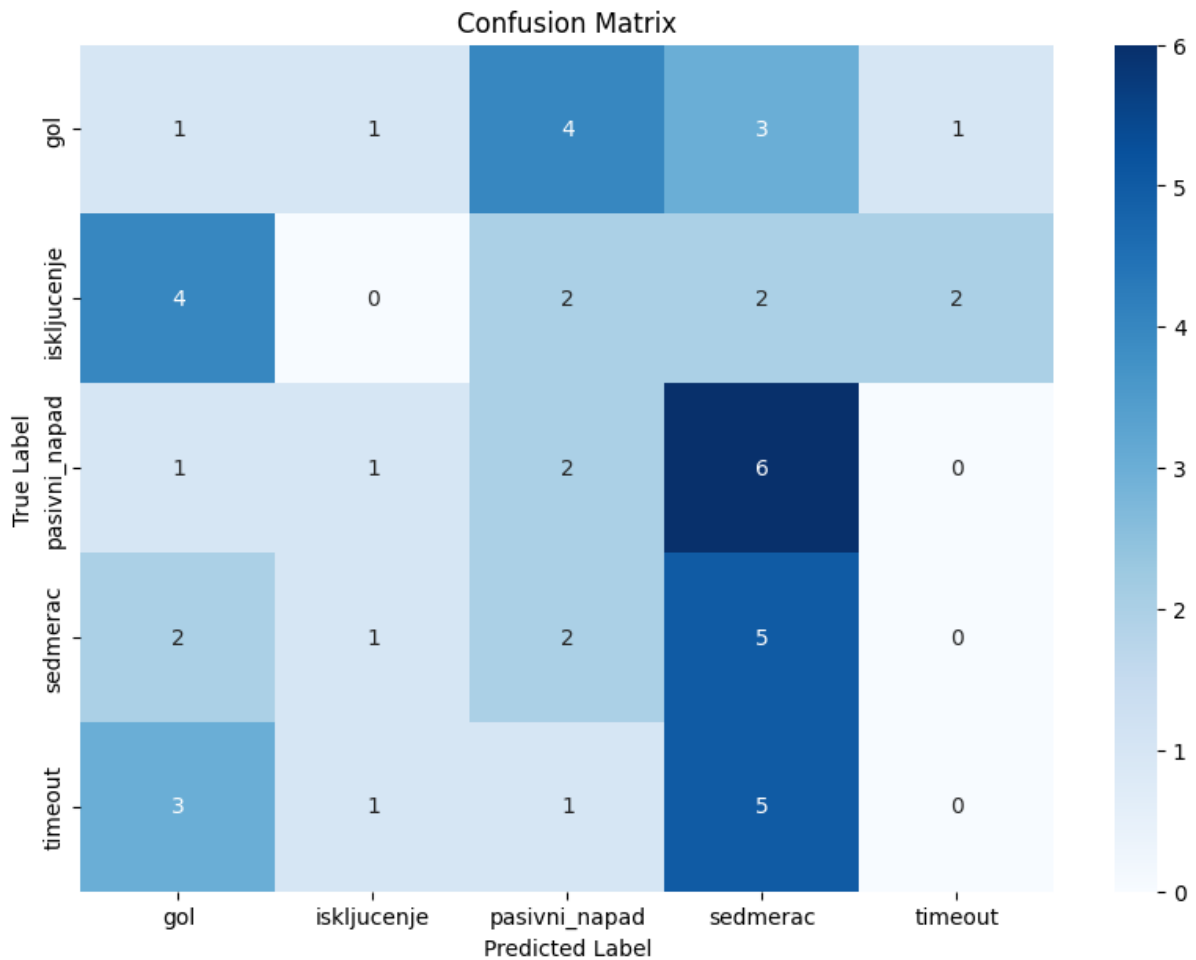
   gol           0.09      0.10      0.10         10
  isključenje    0.00      0.00      0.00         10
 pasivni_napad   0.18      0.20      0.19         10
   sedmerac     0.24      0.50      0.32         10
   timeout      0.00      0.00      0.00         10

 accuracy                0.16         50
 macro avg              0.10      0.16      0.12         50
 weighted avg           0.10      0.16      0.12         50

```

Slika 12. Metrike evaluacije modela MobileNet

Na slici 13, gdje je prikazana matrica konfuzije modela MobileNet, vidimo da isto kao i model VGG16 točno predviđa klasu „sedmerac“. Za razliku od prethodnih modela, prvi puta nam se pojavljuje broj 0 na glavnoj dijagonali. To nam govori da model, u ovom slučaju za klasu „isključenje“ i „timeout“, nije niti jedanput točno predvidio. Dakle, nama je u cilju da vrijednosti na dijagonali budu što veće, što ukazuje na bolji rad modela.



Slika 13. Matrica konfuzije modela MobileNet

8. Optimizacija modela

Optimizacija modela konvolucijske neuronske mreže (CNN) je ključni korak u poboljšanju njegove točnosti i učinkovitosti u prepoznavanju obrazaca, poput sudačkih gesti u rukometu. Prvi korak u optimizaciji uključuje analizu i proširenje skupa podataka; veći i raznovrsniji skup podataka može poboljšati sposobnost modela da generalizira i pravilno prepozna različite geste. Uvođenje tehnika poput „data augmentation“ (npr. rotacija, promjena veličine, osvjetljenja) može značajno povećati raznolikost podataka i pomoći modelu da bolje nauči ključne značajke. Drugi ključni korak je „tuning hiperparametara“, uključujući broj i veličinu konvolucijskih filtara, dubinu mreže, veličinu batch-a i stopu učenja. Metode poput „grid search“ i „random search“ mogu se koristiti za ispitivanje optimalnih vrijednosti ovih hiperparametara.

Uvođenje tehnika regularizacije kao što su „dropout“ i „batch normalization“ može pomoći u smanjenju pretreniranja i poboljšanju stabilnosti modela. Evaluacija rezultata pomoću „cross-validation“ pomaže u procjeni robusnosti modela na različitim podskupovima podataka i smanjuje mogućnost „overfittinga“.

Na kraju, praćenje učenja kroz „learning curves“ omogućuje identificiranje problema poput pretreniranja ili nedovoljno treniranja i omogućava pravovremene korekcije. Implementacija ovih strategija optimizacije omogućit će postizanje bolje točnosti modela, omogućujući mu da preciznije prepozna različite sudačke geste i time doprinese većoj preciznosti u sportskim analizama.

Svaki model funkcionira na različiti način, zbog toga je potrebno svakom modelu pristupiti posebno. U ovom radu, optimizaciju svakog modela vršimo na način da isprobavamo različite postavke poput veličina skupa podataka, dimenzija slika, broj epoha, veličinu „batch-a“, te broj slojeva.

8.1. Optimizacija klasičnog CNN-a

Kako bi mogli optimizirati model potrebno je isprobati različite konfiguracije. Na ovom modelu isprobani su „batch-ovi“ različitih veličina, od 16 do 64 dok se broj epoha kretao od 10 do 25. Isto tako sa veličinom dimenzija slike, od preporučene veličine 224x224, sve do 100x300. Dodavanje sve do 6 konvolucijskih slojeva činilo je modelom pre kompleksnim za naš skup podataka a smanjivanje broja slojeva do 2 činilo je model pre jednostavnim da prepozna neke zahtjevnije obrasce na slikama. Neke od rezultata testiranja možemo vidjeti u tablici 1.

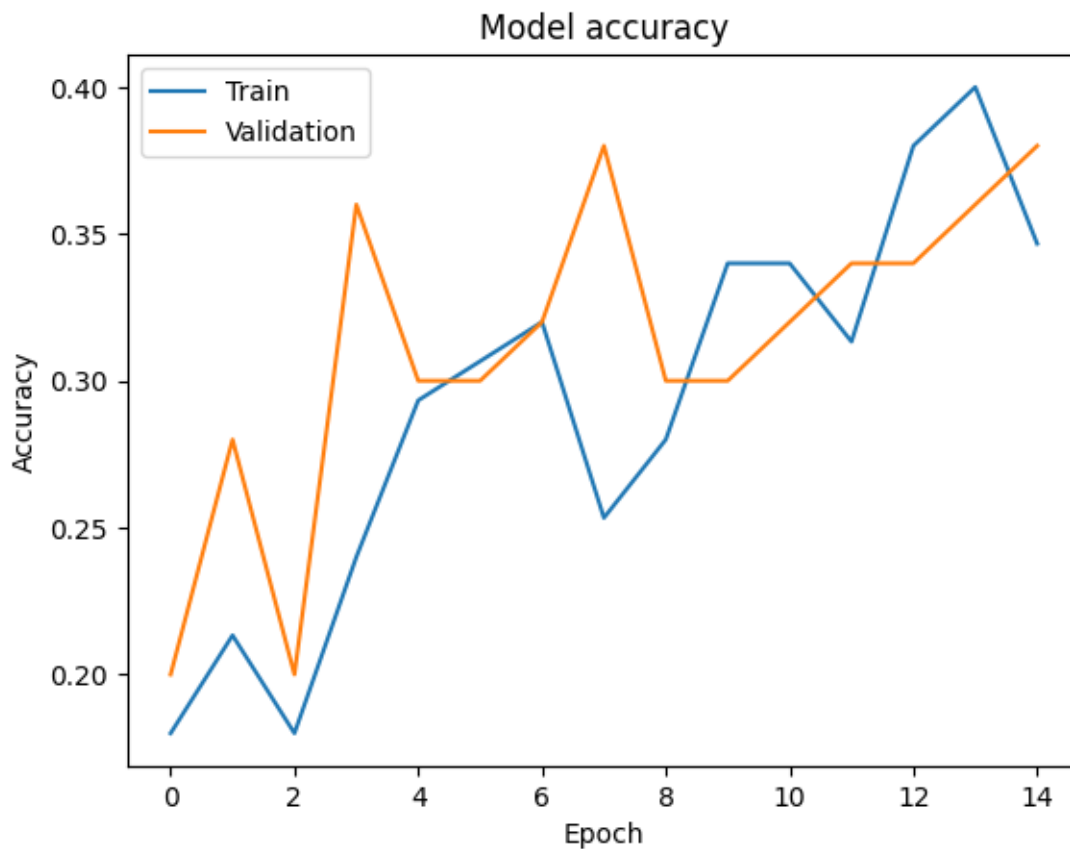
Tablica 1. Rezultati različitih postavka CNN-a

Batch size	Epohe	Dimenzija	Konvolucijski slojevi	Accuracy	Recall	F1-score
64	25	100x300	2	0.36	0.16	0.15
32	10	224x224	4	0.46	0.22	0.16
16	15	224x224	6	0.20	0.20	0.06

Za navedeni model jedne postavke koje su pružale najviše uspjeha su:

- „Batch size“:32
- Broj epoha : 15
- Dimenzije slika: 150x300
- 4 konvolucijska sloja
- 4 „MaxPooling“ sloja
- 1 „Flatten “ sloj
- 2 „Dense“ sloja
- 1 „Dropout“ sloj

Na slici 14 vidimo graf koji nam prikazuje kako se model ponaša kroz epohe. Vidimo da model fluktuirá kroz različite epohe te da pri kraju kreće ponovno padati, što znači da ako bi stavili više od 15 epoha model se ne bi poboljšao.



Slika 14. Graf treniranja modela CNN

Rezultati poboljšanog modela nalaze se na slici 15. Rezultati testiranja modela pokazuju ukupnu točnost od 54%, što predstavlja poboljšanje u odnosu na prethodnu evaluaciju. Međutim, dublja analiza klasifikacijskog izvještaja otkriva značajne varijacije u učinkovitosti modela na pojedinačnim klasama.

- **Gol:** Model je postigao preciznost od 0.28 i „recall“ od 0.50, što znači da je 28% predviđenih "golova" točno klasificirano, dok je prepoznato 50% stvarnih golova. F1-score iznosi 0.36, što ukazuje na umjerenu učinkovitost u prepoznavanju ove geste.
- **Isključenje:** Model postiže preciznost od 0.31 i „recall“ od 0.40, s F1-scoreom od 0.35. To sugerira da je model donekle uspješan u prepoznavanju isključenja, iako postoji prostor za poboljšanje.
- **Pasivni napad:** Preciznost za ovu klasu iznosi 0.11, dok je „recall“ 0.10. F1-score je vrlo nizak, samo 0.11, što ukazuje na lošu učinkovitost modela u prepoznavanju pasivnog napada, te da model često promašuje ovu gestu.
- **Sedmerac:** Ova klasa ima izuzetno visoku preciznost od 1.00, što znači da su svi predviđeni "sedmerci" točno klasificirani, ali „recall“ iznosi samo 0.20, što sugerira da model prepoznaje samo 20% stvarnih sedmeraca. F1-score od 0.33 ukazuje na neravnotežu između preciznosti i „recalla“.

- **Timeout:** Model postiže preciznost od 0.25 i „recall“ od 0.20, s F1-scoreom od 0.22, što sugerira nisku učinkovitost u prepoznavanju ove geste, slično kao i kod drugih klasa.

Prosječne metričke vrijednosti (macro avg) pokazuju preciznost od 0.39, „recall“ od 0.28 i F1-score od 0.27, dok su ponderirane prosječne vrijednosti (weighted avg) iste. Ovi rezultati ukazuju na to da model ima određeni napredak u prepoznavanju sudačkih gesti, no još uvijek postoji značajan prostor za poboljšanje, posebno u pogledu ravnoteže između preciznosti i „recalla“ za sve klase. Daljnja optimizacija modela i povećanje raznolikosti skupa podataka mogla bi pomoći u poboljšanju performansi modela.

```

Test accuracy: 0.5400000214576721
2/2 ----- 0s 111ms/step
      precision    recall  f1-score   support

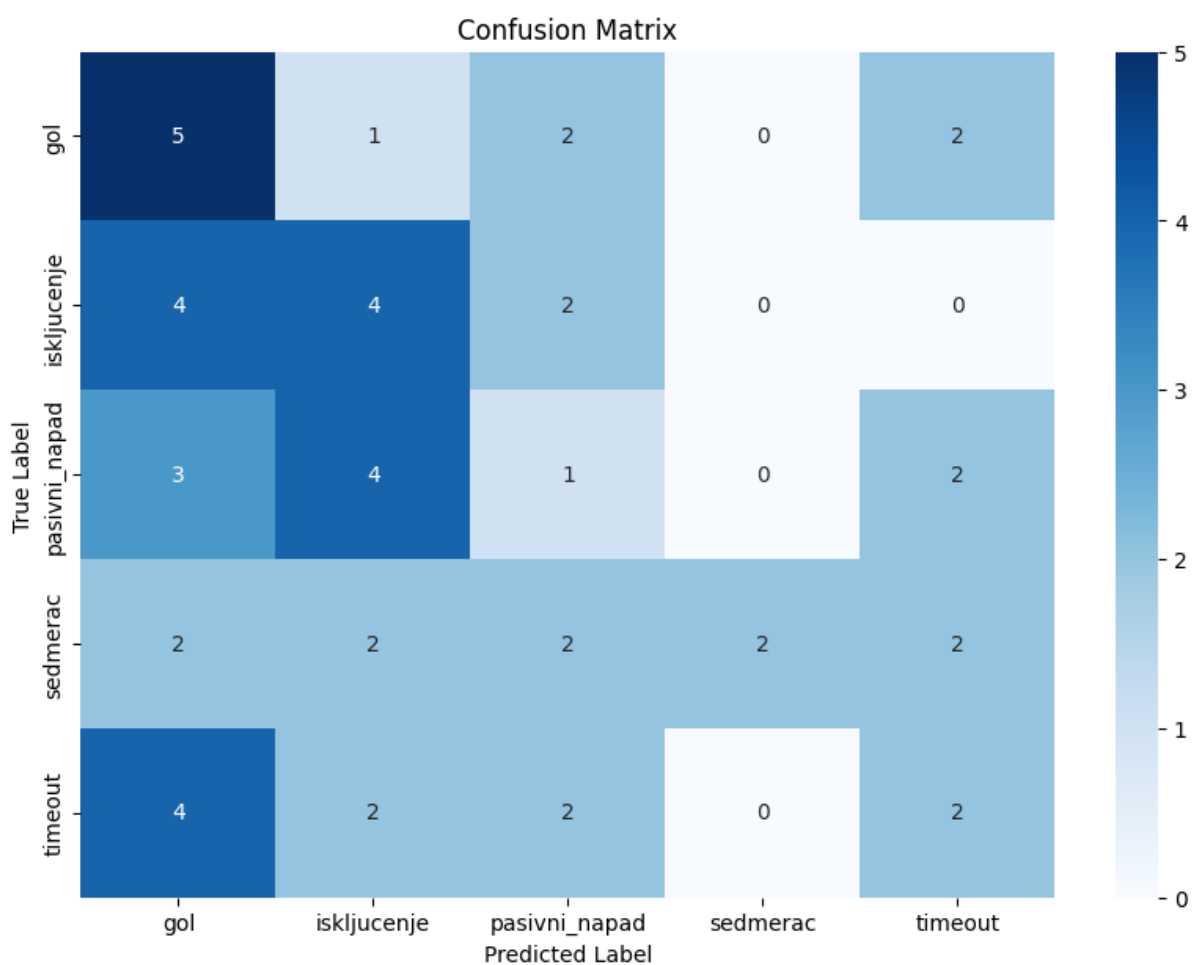
   gol           0.28      0.50      0.36         10
  iskljucenje    0.31      0.40      0.35         10
 pasivni_napad   0.11      0.10      0.11         10
   sedmerac     1.00      0.20      0.33         10
   timeout      0.25      0.20      0.22         10

 accuracy              0.28         50
 macro avg             0.39      0.28      0.27         50
 weighted avg          0.39      0.28      0.27         50

```

Slika 15. Metrike evaluacije modela CNN nakon optimizacije

Matrica konfuzije optimiziranog modela (slika 16.) prikazuje da klasa „sedmerac“ je jedina koja ima samo dvije točno predviđene instance i ostale ima 0. Klasa „gol“ ima najviši broj točno predviđenih instanci ali zato ima najviše instanci koje se miješaju sa drugima.



Slika 16. Matrica konfuzije modela CNN nakon optimizacije

8.2. Optimizacija modela VGG16

Na sličan način optimiziramo model VGG16 kao i klasični CNN. Tablica 2. prikazuje par različitih rezultata dobivenih kombinacijom različitih postavka.

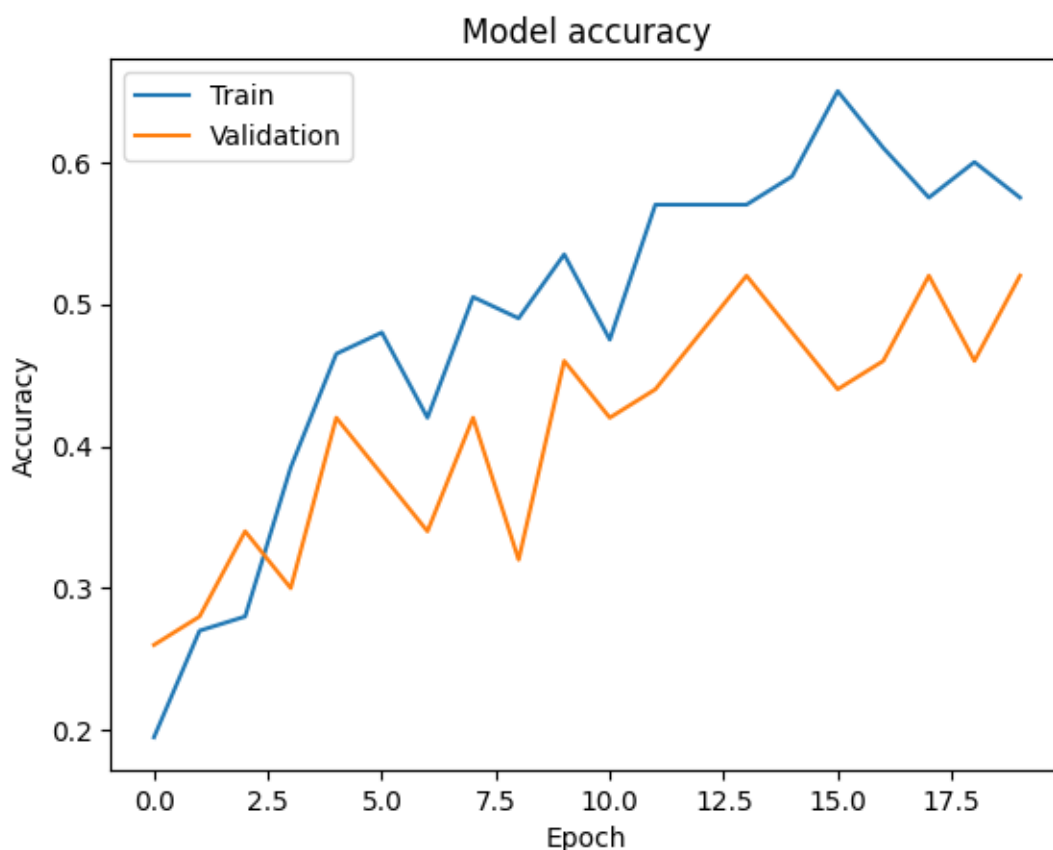
Tablica 2. Rezultati različitih postavka VGG16

Batch size	Epohe	Dimenzija	Dodatni slojevi	Accuracy	Recall	F1-score
16	20	224x224	1	0.16	0.16	0.15
64	15	224x224	3	0.14	0.14	0.13
32	10	224x224	3	0.12	0.12	0.10

Isprobavanje različitih postavka dolazimo do onih koje daju najbolji rezultat.

- „Batch size“:32
- Broj epoha: 15
- Dimenzije slika: 224x224
- 18 baznih slojeva iz VGG16
- 1 „GlobalAveragePooling2D“ sloj
- 1 „Dense(1024)“ sloj
- 1 „Desnse(output)“ sloj

Graf na slici 17 prikazuje kako se modelu kroz treniranje preciznost povećava sve do 15. epohe gdje nakon toga krene naglo padati , što ukazuje da je optimalan broj epoha upravo 15.



Slika 17. Graf treniranja modela VGG16

Rezultati testiranja poboljšanog modela pokazuje nešto bolje rezultate (slika 18.). Pokazuju ukupnu točnost od 22%, što ukazuje na to da model ispravno klasificira samo 22% sudačkih gesti iz testnog skupa. Analiza detaljnih rezultata za svaku klasu daje sljedeće uvide:

- **Goal:** Model postiže preciznost od 0.29 i „recall“ od 0.20, što znači da je 29% predviđenih "golova" točno identificirano, dok je samo 20% stvarnih golova

prepoznato. F1-score za ovu klasu iznosi 0.24, što sugerira umjerenu, ali nedovoljnu učinkovitost u prepoznavanju golova.

- **Isključenje:** Klasa isključenja pokazuje bolju izvedbu s preciznošću od 0.33 i „recallom“ od 0.40. F1-score iznosi 0.36, što ukazuje na solidnu sposobnost modela u prepoznavanju ove geste, iako i dalje postoji prostor za poboljšanje.
- **Pasivni napad:** Preciznost za ovu klasu je vrlo niska, samo 0.10, dok je „recall“ također niskih 0.10. F1-score od 0.10 ukazuje na vrlo nisku učinkovitost modela u prepoznavanju pasivnog napada.
- **Sedmerac:** Model postiže preciznost od 0.29 i „recall“ od 0.20, s F1-scoreom od 0.24, što pokazuje da model ima umjerene poteškoće u prepoznavanju sedmeraca, slično kao i kod golova.
- **Timeout:** Preciznost za timeout je 0.14, dok je „recall“ 0.20, što daje F1-score od 0.17. Ovi rezultati sugeriraju da model ima slične probleme u prepoznavanju timeout gesti kao i kod ostalih klasa.

Prosječne metričke vrijednosti (macro avg) su 0.23 za preciznost, 0.22 za recall i 0.22 za F1-score, dok su ponderirane (weighted avg) metričke vrijednosti iste. Ovi rezultati ukazuju na to da model ima konstantne, ali niske performanse u prepoznavanju svih klasa sudačkih gesti. Poboljšanje performansi modela vjerojatno će zahtijevati dodatnu optimizaciju, uključujući proširenje skupa podataka, fino podešavanje hiperparametara, te primjenu naprednijih tehnika učenja.

```
Accuracy: 0.22
Recall: 0.22
F1 Score: 0.2201782531194296

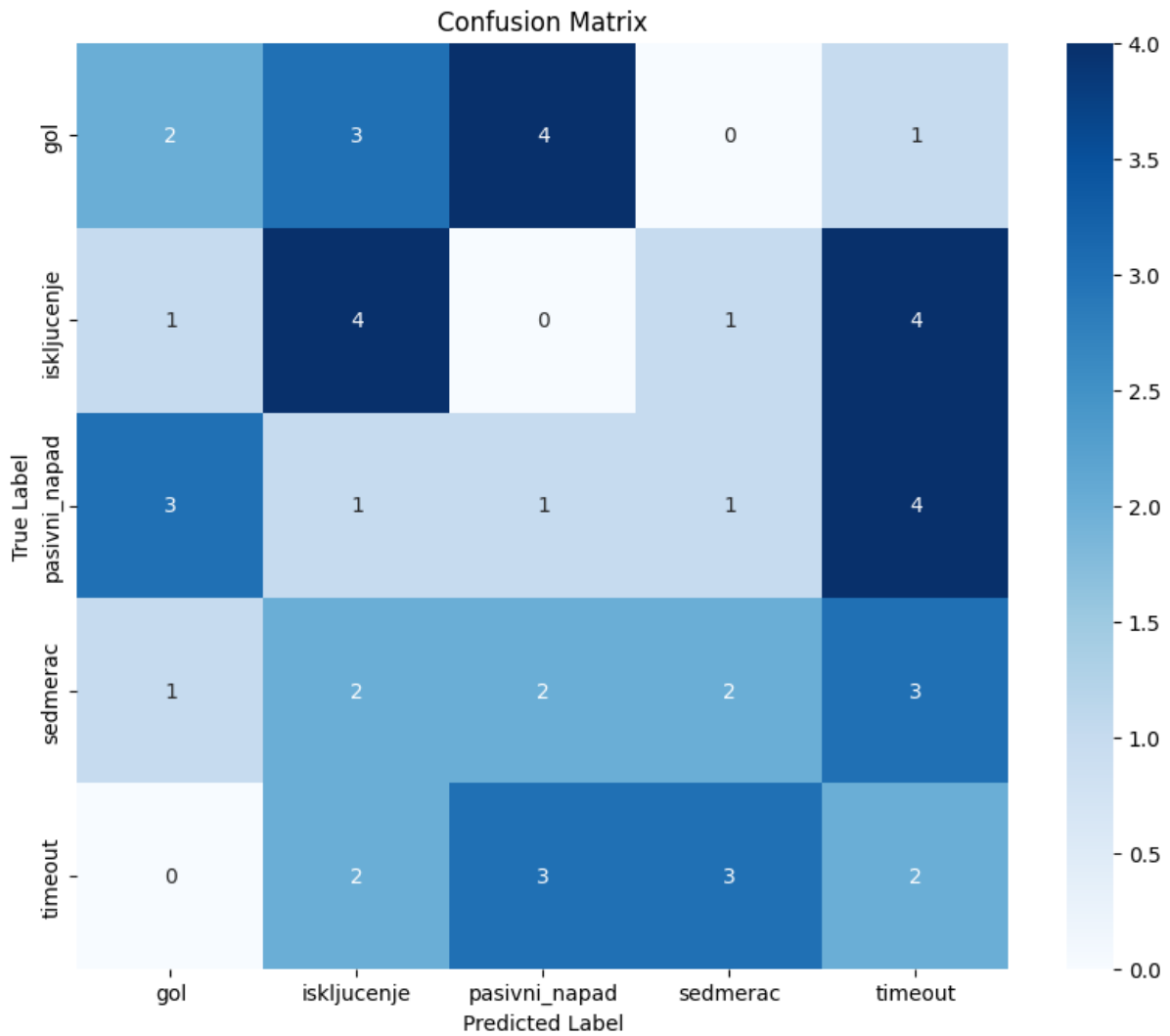
Classification Report:

```

	precision	recall	f1-score	support
gol	0.29	0.20	0.24	10
isključenje	0.33	0.40	0.36	10
pasivni_napad	0.10	0.10	0.10	10
sedmerac	0.29	0.20	0.24	10
timeout	0.14	0.20	0.17	10
accuracy			0.22	50
macro avg	0.23	0.22	0.22	50
weighted avg	0.23	0.22	0.22	50

Slika 18. Metrike evaluacije modela VGG16 nakon optimizacije

Na slici 19. , koja prikazuje novu matricu konfuzije vidimo da je ovoga puta klasa „isključenje“ ta sa najviše točno predviđenim instanci. Također vidimo poprilično visoke brojeve izvan dijagonale , što ukazuje da model često predviđa pogrešne klase.



Slika 19. Matrica konfuzije modela VGG16 nakon optimizacije

8.3. Poboljšanje modela MobileNet

Kao i na prijašnjim tablicama, na tablici 3. možemo vidjeti različite rezultate modela.

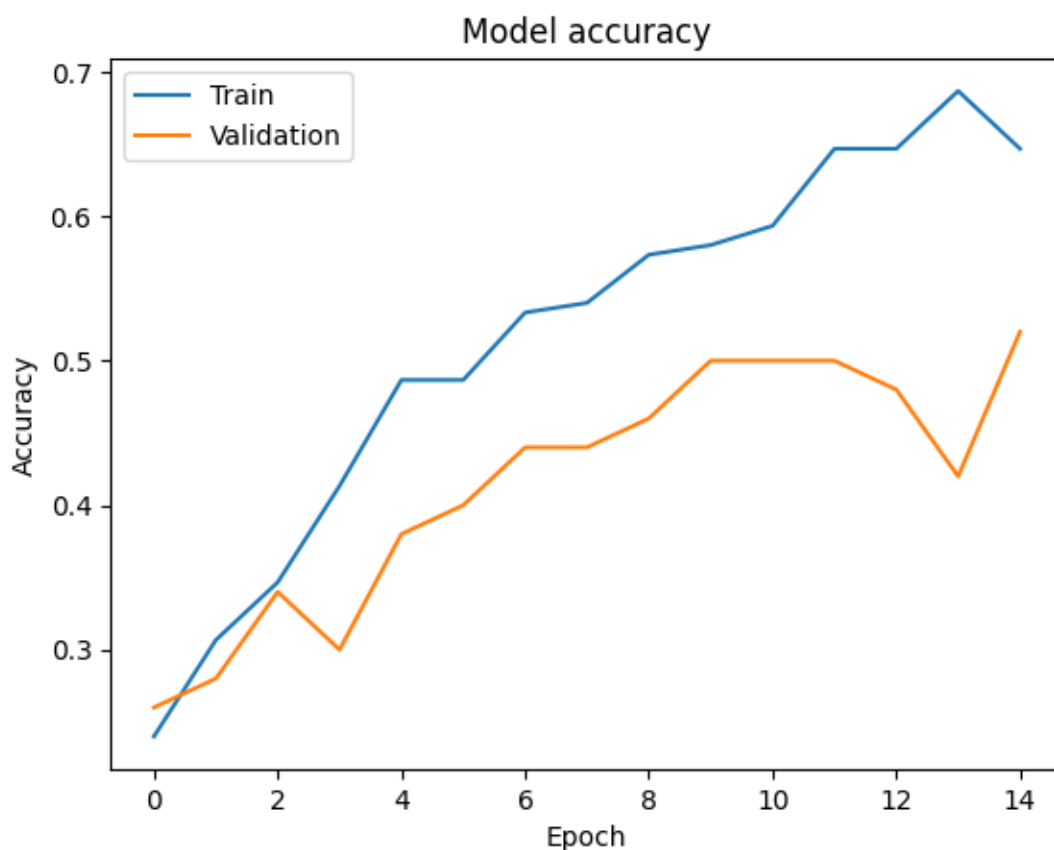
Tablica 3. Rezultati različitih postavka MobileNet-a

Batch size	Epohe	Dimenzija	Dodatni slojevi	Accuracy	Recall	F1-score
32	10	100x300	4	0.10	0.16	0.12
64	20	150x300	4	0.18	0.18	0.18
16	15	224x224	2	0.15	0.18	0.16

Od svih navedenih modela MobileNet ima najveću ukupno tačnost od 58%. Postavke koje k tome doprinose su sljedeće :

- „Batch size“: 32
- Broj epoha: 13
- Dimenzije slika: 224x224
- 53 baznih slojeva MobileNeta-a
- 1 GlobalAveragePooling2d sloj
- 1 „Dense (128)“ sloj
- 1 „Dropout“ sloj
- 1 „Dense(output)“ sloj

Graf treniranja modela MobileNet sa slike 20. prikazuje kako tačnost modela raste sve do otprilike 13. epohe što se pokazalo kao najbolji rezultat.



Slika 20. Graf treniranja modela MobileNet

Kada analiziramo rezultate modela MobileNet sa ukupnom točnošću od 58% (slika 21.), primjećujemo nekoliko ključnih aspekata koji ukazuju na učinkovitost modela u prepoznavanju sudačkih gesti:

- **Gol:** Model ima preciznost od 0.22 i „recall“ od 0.20 za ovu klasu, što znači da je 22% predviđenih "golova" točno identificirano, dok je 20% stvarnih golova prepoznato. F1-score za golove iznosi 0.21, što pokazuje da model ima skromne performanse u prepoznavanju golova.
- **Isključenje:** Slični rezultati vrijede i za klasu isključenja, gdje preciznost i „recall“ iznose 0.22 i 0.20, respektivno. F1-score od 0.21 ukazuje na nisku učinkovitost u prepoznavanju isključenja.
- **Pasivni napad:** Za ovu klasu, model pokazuje vrlo nisku preciznost od 0.07 i „recall“ od 0.10, s F1-scoreom od 0.08. Ovi rezultati sugeriraju značajne poteškoće u prepoznavanju pasivnog napada.
- **Sedmerac:** Klasa sedmeraca također pokazuje nisku preciznost od 0.07 i „recall“ od 0.10, s F1-scoreom od 0.08, što ukazuje na slične poteškoće u prepoznavanju ove geste.
- **Timeout:** Preciznost za timeout iznosi 0.25, dok je „recall“ 0.10, što daje F1-score od 0.14. Ovi rezultati pokazuju da model ima ograničenu sposobnost prepoznavanja timeout gesti.

Prosječne metričke vrijednosti (macro avg) su 0.17 za preciznost, 0.14 za „recall“ i 0.15 za F1-score. Ponderirane (weighted avg) vrijednosti su slične, što dodatno potvrđuje da model pokazuje slabu učinkovitost u prepoznavanju svih klasa.

Unatoč ukupnoj točnosti od 58%, što može izgledati kao umjereno prihvatljivo, detaljna analiza rezultata otkriva da model ne uspijeva ujednačeno prepoznati sve klase sudačkih gesti. Ova situacija sugerira da model možda ne generalizira dobro na testnim podacima.

```

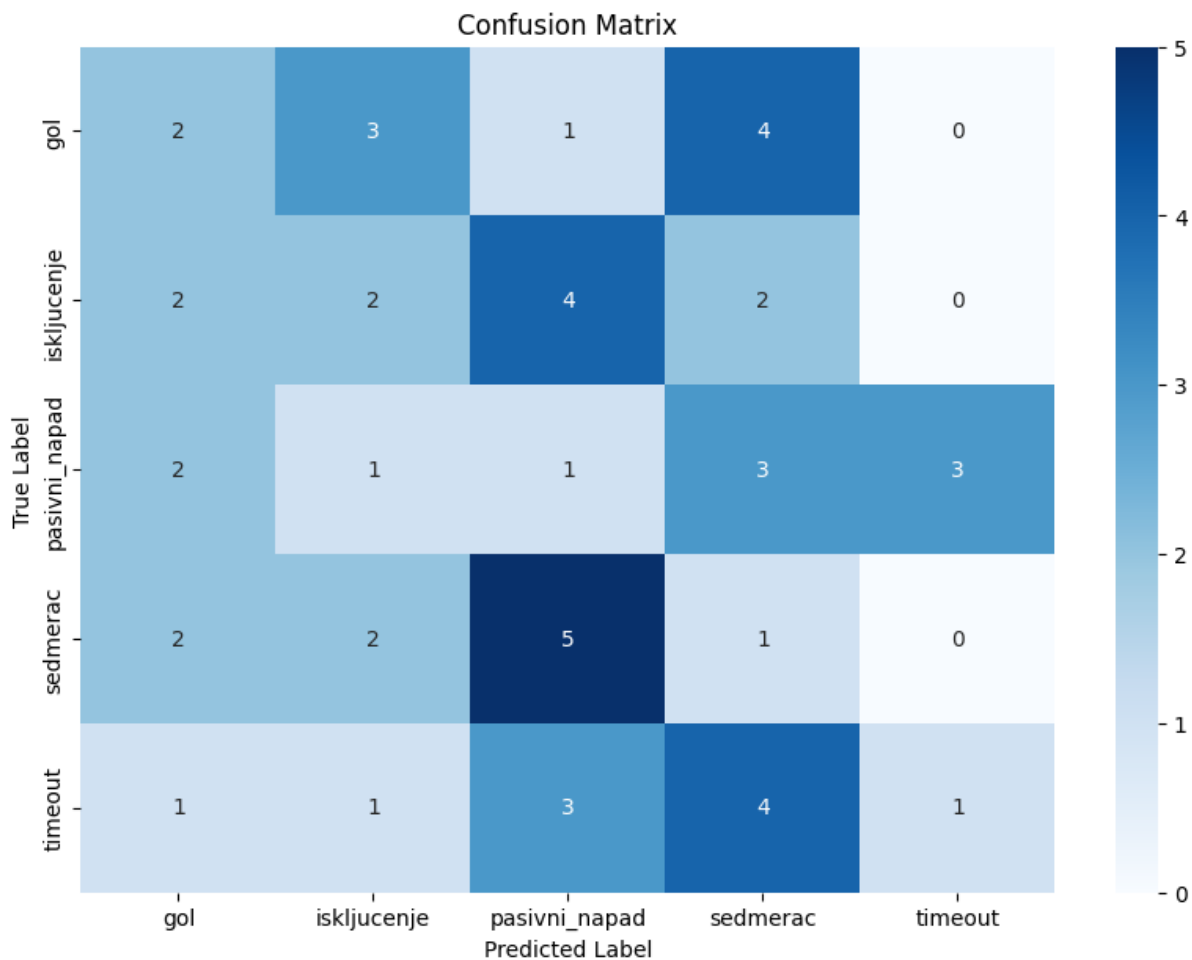
Test accuracy: 0.5799999833106995
2/2 ————— 3s 1s/step

```

	precision	recall	f1-score	support
gol	0.22	0.20	0.21	10
isključenje	0.22	0.20	0.21	10
pasivni_napad	0.07	0.10	0.08	10
sedmerac	0.07	0.10	0.08	10
timeout	0.25	0.10	0.14	10
accuracy			0.14	50
macro avg	0.17	0.14	0.15	50
weighted avg	0.17	0.14	0.15	50

Slika 21. Metrike evaluacije modela MobileNet nakon optimizacije

Slika matrice konfuzije (slika 22.) , prikazuje vrijednosti od 1 do 2 na dijagonali što znači da je model svaku od klasa jednom do dva puta točno predvidio. Taj podatak nam govori da je model na dobrom putu ali mu je potreba daljnja optimizacija kako bi postigao bolje rezultate.



Slika 22. Matrica konfuzije nakon optimizacije modela MobileNet

9. YOLO

YOLO (You Only Look Once) je napredna tehnika detekcije objekata koja se temelji na dubokom učenju i prvi put je predstavljena 2015. godine od strane Joseph Redmona i njegovih kolega. Ova metoda se izdvaja po svojoj sposobnosti da simultano detektira više objekata unutar jedne slike ili video sekvence, omogućavajući brzu i efikasnu analizu slika u stvarnom vremenu. YOLO koristi arhitekturu konvolucijskih neuronskih mreža (CNN) koja obrađuje cijelu sliku odjednom, dijeleći je na mrežu ćelija (najčešće 7x7 ili 13x13), pri čemu svaka ćelija predviđa okvire (bounding boxes) i vjerojatnosti pripadnosti određenim klasama objekata. Ovaj pristup značajno ubrzava proces detekcije, jer umjesto višekratnog prolaska kroz sliku, kao što to rade tradicionalne metode, YOLO analizira sve u jednom prolazu.

Tokom godina, YOLO je evoluirao kroz više verzija, svaka s novim poboljšanjima. YOLOv2 je uveo nove tehnike poput višestrukih skala i unaprijed treninga, što je povećalo točnost detekcija. YOLOv3 je dodao složenije slojeve za bolje prepoznavanje objekata, dok su YOLOv4 i YOLOv5 optimizirali performanse putem tehnika poput transfer učenja,

augmentacije podataka i poboljšanih strategija za obučavanje, uključujući korištenje novih arhitektura poput CSPDarknet i EfficientDet. Ove verzije su se pokazale izuzetno efikasnim u raznim scenarijima, uključujući one s niskom svjetlošću i kompleksnim scenama s preklapanjem objekata.

Primjene YOLO-a su raznovrsne i obuhvaćaju mnoge industrije. U autonomnim vozilima, YOLO omogućava brzo prepoznavanje drugih vozila, pješaka i prepreka, čime se poboljšava sigurnost na cesti. U video nadzoru, ova tehnika pomaže u identifikaciji sumnjivih aktivnosti u stvarnom vremenu, što može biti ključno za prevenciju kriminala. Također se koristi u robotici, gdje pomaže robotima da se orijentiraju i reagiraju s okruženjem, kao i u raznim aplikacijama za analizu slika i video sadržaja, poput prepoznavanja lica i klasifikacije scena.

Zbog svoje brzine i preciznosti, YOLO je stekao široku popularnost među istraživačima i inženjerima, postajući nezaobilazni alat u području računalnog vida. Njegova jednostavnost korištenja, uz otvoreni pristup i dostupnost predtreniranih modela, omogućava istraživačima da brzo implementiraju i testiraju svoje ideje, dok kontinuirani razvoj i unapređenja garantiraju da će YOLO ostati relevantan i u budućnosti. U skladu s trendovima u umjetnoj inteligenciji i računalnom vidu, očekuje se daljnje usavršavanje algoritama YOLO-a, čime će se dodatno povećati njegova korisnost i primjenjivost u različitim tehnološkim rješenjima.

9.1. Treniranje modela

Treniranje YOLO modela na proizvoljnom skupu podataka izvodi se na način da se podijeli skup na treniranje i validaciju, najčešće u omjeru 80% : 20%. Sve što je potrebno za treniranje je postavljanje gotove instance jedne od mogućih verzija modela, definiranje putanje skupa podataka, postavljanje broja epoha te veličina slike.

```
from ultralytics import YOLO

model = YOLO('yolov8n.pt')

model.train(data='data.yaml', epochs=50, imgsz=300)
```

9.2. Rezultati

Izveštaj o sažetku modela (slika 23.) nakon treniranja pruža ključne informacije o njegovoj strukturi i performansama. Model se sastoji od 168 slojeva i ima ukupno 3,006,623 parametara, što ukazuje na njegovu složenost i kapacitet za učenje. Tokom evaluacije, model nije imao nijednu gradijentnu informaciju (0 gradients), a njegovo računanje zahtijeva 8.1 GFLOPs, što ukazuje na potrebnu procesorsku snagu za obradu.

U evaluaciji modela, korišteno je 45 slika s ukupno 45 instanci objekata. Rezultati su prikazani u formatu koji uključuje preciznost (P), odziv (R) te srednju preciznost (mAP) za različite kriterije:

- **Sve klase:** Model je postigao ukupnu preciznost od 85.1% i odziv od 83.8%, s mAP50 (srednja preciznost na 50% IoU) od 90.1% i mAP50-95 (srednja preciznost na IoU od 50% do 95%) također od 90.1%.

- **Gol:** Model je imao preciznost od 74.4% i vrlo dobar odziv od 96.9%, s mAP50 od 90.8% i mAP50-95 također od 90.8%.
- **Sedmerac:** Ovdje je model ostvario preciznost od 83.1% i odziv od 88.9%, s mAP50 od 88.8% i mAP50-95 također od 88.8%.
- **Timeout:** Ova klasa je imala izuzetno visoku preciznost od 98.7% s odzivom od 77.8%, a mAP50 je iznosio 91.2%, dok je mAP50-95 iznosio 91.2%.
- **Pasivni napad:** Model je postigao preciznost od 79.3% s niskim odzivom od 55.6%, a mAP50 je bio 81.3% i mAP50-95 također 81.3%.
- **Isključenje:** Ova klasa je postigla izvanredne rezultate s preciznošću od 90% i visokim odzivom od 99.7%, a mAP50 i mAP50-95 su bili 98.4%.

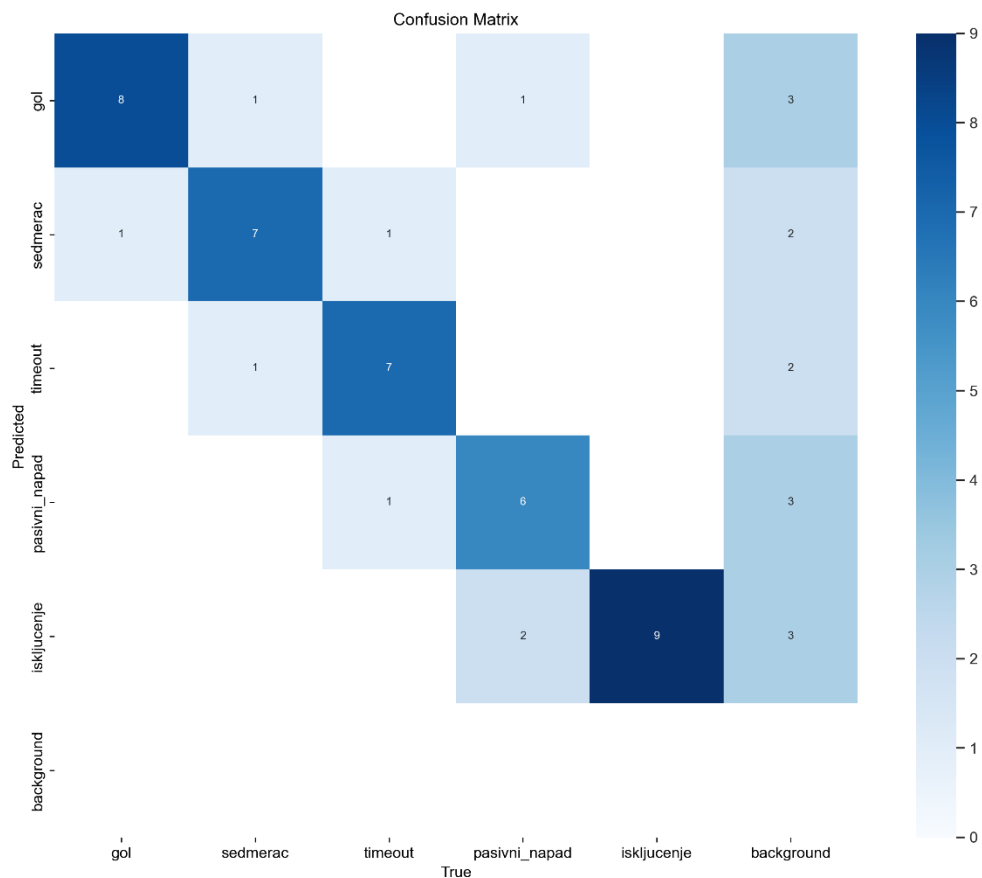
```

Model summary (fused): 168 layers, 3,006,623 parameters, 0 gradients, 8.1 GFLOPs
  Class      Images  Instances  Box(P  R      mAP50  mAP50-95):
    all       45       45         0.851  0.838  0.901  0.901
    gol        9        9          0.744  0.969  0.908  0.908
    sedmerac   9        9          0.831  0.889  0.888  0.888
    timeout    9        9          0.987  0.778  0.912  0.912
    pasivni_napad 9        9          0.793  0.556  0.813  0.813
    iskljucenje 9        9          0.9    0.997  0.984  0.984

```

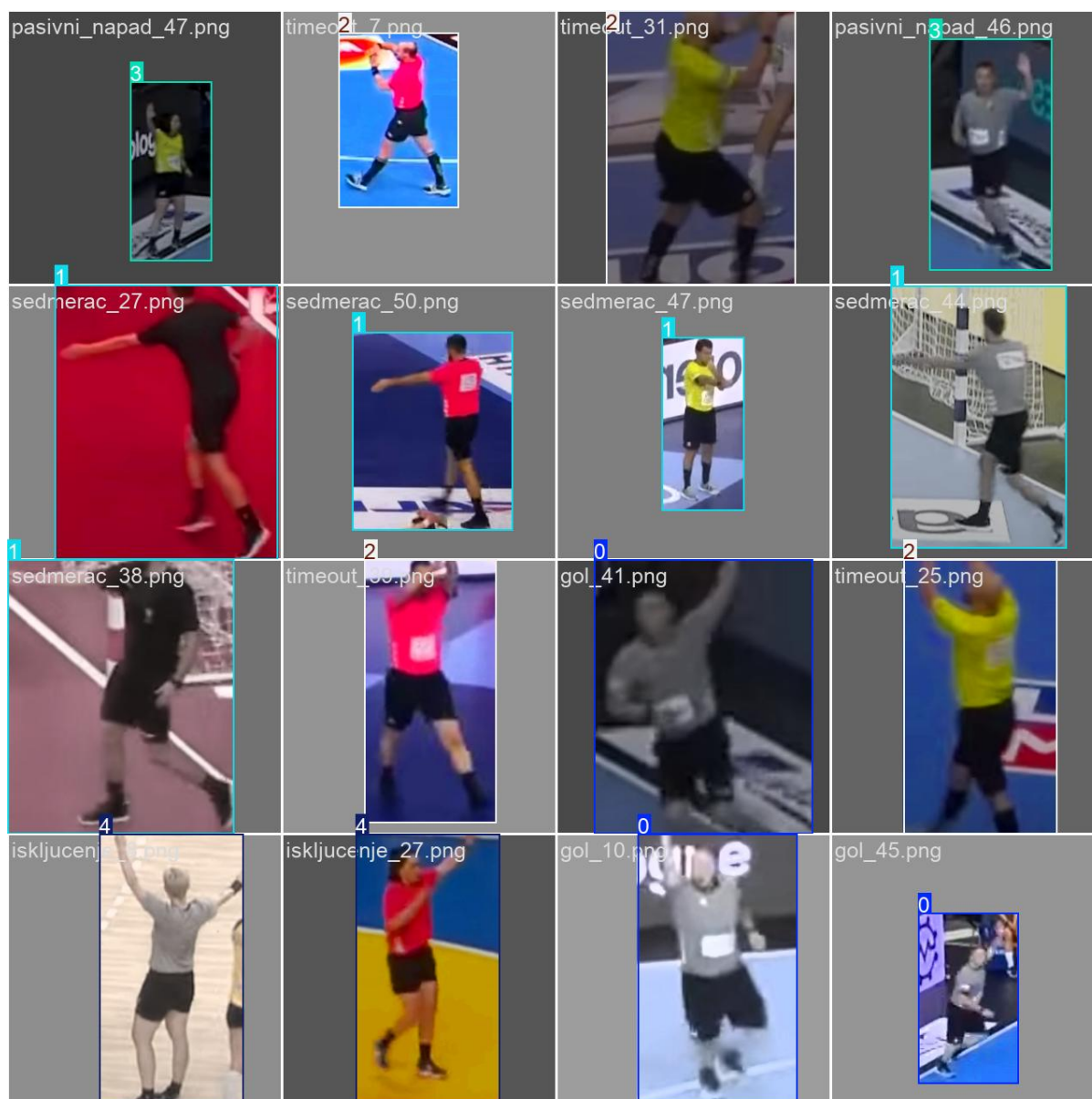
Slika 23. Rezultati modela YOLO

Slika 24. prikazuje matricu konfuzije gdje vidimo iznimno visoke brojeve na glavnoj dijagonali što upućuje da model često predviđa ispravnu klasu.



Slika 24. Matrica konfuzije modela YOLO

Također u sklopu ovog modela možemo vidjeti kako izgledaju „batchevi“ za treniranje. Na slici 25. vidimo različite klase pomiješane te oko njih vidimo „okvire“ (bounding boxes) koji se koriste za označavanje objekta na slici.



Slika 25. "Batch" za treniranje

10. Rasprava

Rezultati eksperimenata provedenih u ovom radu pokazali su da je moguće primijeniti konvolucijske neuronske mreže za prepoznavanje sudačkih gesti u rukometu, ali s različitim stupnjevima uspješnosti. Iako su svi modeli postigli određenu razinu točnosti, postoji značajan prostor za poboljšanje.

Najbolje rezultate postigao je optimizirani model MobileNet, koji je dostigao ukupnu točnost od 58%. Međutim, dublja analiza pokazala je da model ima poteškoća s ujednačenim prepoznavanjem svih klasa gesti, što ukazuje na potrebu za daljnjim usavršavanjem. Ostali modeli, poput klasičnog CNN-a i VGG16, pokazali su nešto niže performanse, ali su također otkrili potencijal za razvoj.

Optimizacijski proces, koji je uključivao podešavanje hiperparametara i eksperimentiranje s različitim arhitekturama modela, bio je ključan za postizanje boljih rezultata. Međutim, rezultati su pokazali da postoji ograničenje u poboljšanju performansi samo kroz optimizaciju.

U kontekstu detekcije sudačkih gesti, YOLO (You Only Look Once) predstavlja dodatnu perspektivu koja bi mogla značajno unaprijediti točnost i brzinu prepoznavanja. Ova metoda detekcije objekata omogućava analizu video sadržaja u stvarnom vremenu, što bi omogućilo brže i preciznije prepoznavanje gesti tijekom utakmica. YOLO koristi arhitekturu konvolucijskih mreža koja obrađuje cijelu sliku odjednom, čime se eliminira potreba za višekratnim prolazima kroz sliku, što rezultira bržim procesom prepoznavanja. Integracijom YOLO-a u sustav prepoznavanja gesti, postoji mogućnost poboljšanja performansi modela, posebno u dinamičnim situacijama kao što su sportske utakmice, gdje je važno odmah reagirati na promjene u sceni. Iako je potrebno dodatno istražiti i prilagoditi ovu tehniku za specifične potrebe prepoznavanja gesti u rukometu, YOLO predstavlja obećavajuću alternativu koja bi mogla donijeti značajna poboljšanja u preciznosti i efikasnosti detekcije.

11. Nadograđivanje sustava

Kvaliteta i veličina skupa podataka mogu imati značajan utjecaj na performanse modela. Korištenje većeg i raznovrsnijeg skupa podataka, koji bi uključivao različite snimke s različitim kamera i pod različitim uvjetima osvjetljenja, moglo bi poboljšati generalizacijsku sposobnost modela. Složenost sudačkih gesti i sličnost nekih od njih predstavlja značajan izazov za modele dubokog učenja. Razvijanje sofisticiranijih arhitektura modela i primjena tehnika pažnje mogle bi pomoći u boljem razlikovanju sličnih gesti.

Budući rad bi se trebao usredotočiti na rješavanje ovih ograničenja. To bi uključivalo prikupljanje većeg i kvalitetnijeg skupa podataka, razvijanje novih arhitektura modela specifično dizajniranih za prepoznavanje sudačkih gesti, te primjenu naprednih tehnika dubokog učenja, kao što su generativni modeli i tehnike pojačanog učenja. Također, zanimljivo bi bilo istražiti primjenu ovih modela u realnom vremenu, što bi omogućilo njihovu integraciju u sportske prijenose i analizu.

Isto tako, ovaj sustav u skladu sa sustavom koji može u stvarnom vremenu prepoznati ispravnu odluku bez upotrebe sudaca, pridonio bi revolucionarnom spajanju tehnologije i sporta.

12. Zaključak

Korištenjem konvolucijskih neuronskih mreža (CNN), razvijeni su modeli koji su pokazali sposobnost prepoznavanja različitih sudačkih signala, iako s varirajućim stupnjem uspješnosti. Istraživanje je pokazalo da je primjena CNN modela izvediva, čime se otvaraju nove mogućnosti za primjenu u sportskoj analitici. Optimizacija hiperparametara, kao što su veličina batch-a, broj epoha i arhitektura mreže, značajno je utjecala na performanse modela. Također, kvaliteta i količina podataka za treniranje pokazali su se ključnim za postizanje veće točnosti

u prepoznavanju sudačkih gesti. Iako su postignuti značajni rezultati, modeli još uvijek pokazuju ograničenja u generalizaciji na različite uvjete snimanja i varijacije u izvođenju gesti.

U budućnosti, daljnji napredak može se postići proširenjem skupa podataka, uključivanjem raznolikih snimaka s različitih kamera i pod različitim uvjetima osvjetljenja, što bi moglo poboljšati generalizacijsku sposobnost modela. Razvoj sofisticiranijih arhitektura, kao što su modeli temeljeni na pažnji, mogao bi dodatno unaprijediti prepoznavanje prostorno-vremenskih obrazaca u videu. Također, primjena tehnika pojačanog učenja mogla bi omogućiti modelima da se prilagođavaju dinamičnim promjenama tijekom igre, dok bi integracija s drugim tehnologijama, poput senzora za praćenje pokreta, poboljšala točnost prepoznavanja. Razvoj sustava koji radi u stvarnom vremenu omogućio bi trenutnu analizu sudačkih odluka tijekom utakmice, što bi dodatno povećalo korisnost ovih modela.

Zaključno, istraživanje predstavlja značajan doprinos automatiziranoj analizi sportskih događaja. Iako su trenutačni modeli još uvijek u razvoju, njihove potencijalne primjene su brojne, od poboljšanja kvalitete suđenja do analize performansi igrača i timova te kreiranja alata za treniranje i edukaciju. Daljnji razvoj ove tehnologije može značajno unaprijediti sportsku analitiku i transparentnost suđenja, otvarajući nove mogućnosti za istraživanje i komercijalizaciju. U tom kontekstu, važno je također razmotriti etičke aspekte, uključujući autonomno odlučivanje i potencijalne pristranosti u modelima, kao i utjecaj ove tehnologije na same sportske uloge, posebno uloge sudaca, te na odnose unutar sportske zajednice.

Literatura

- [1] 1 IBM, "Computer Vision," *IBM Topics*. [Online]. Dostupno: <https://www.ibm.com/topics/computer-vision>. [Pristupljeno: 31. kolovoza 2024].
- [2] 222 IBM, "Machine Learning," *IBM Topics*. [Online]. Dostupno: <https://www.ibm.com/topics/machine-learning>. [Pristupljeno: 31. kolovoza 2024].
- [3] 333 I. Goodfellow, Y. Bengio, i A. Courville, *Deep Learning*. Cambridge, MA, SAD: MIT Press, 2016. [Online]. Dostupno: <http://www.deeplearningbook.org/>. [Pristupljeno: 31. kolovoza 2024].
- [4] 444 C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, SAD: Springer, 2006.
- [5] 555 Y. LeCun, Y. Bengio, i G. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, str. 2278-2324, Nov. 1998.

- [6] 666 A. Krizhevsky, I. Sutskever, i G. H. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," u *Advances in Neural Information Processing Systems*, vol. 25, 2012, str. 1097-1105.
- [7] 777 D. P. Kingma i M. Welling, "Auto-Encoding Variational Bayes," u *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. [Online]. Dostupno: <https://arxiv.org/abs/1312.6114>. [Pristupljeno: 31. kolovoza 2024].
- [8] 888 T. Mikolov, K. Chen, G. S. Corrado, i J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," u *Proceedings of the 26th International Conference on Neural Information Processing Systems*, 2013, str. 3111-3119.
- [9] 999 A. Y. Ng, "Feature Selection, L1 vs. L2 Regularization, and Rotational Invariance," *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004, str. 78.
- [10] 101010 I. J. Goodfellow, "Deep Learning for Computer Vision," *University of Toronto*, Toronto, ON, Kanada, 2013. [Online]. Dostupno: <https://www.cs.toronto.edu/~kriz/learning-features-2013.pdf>. [Pristupljeno: 31. kolovoza 2024].
- [11] 111111 B. R. Achuthan, A. V. M., i A. A. V. S., "A Survey on Deep Learning and Its Applications," *International Journal of Computer Applications*, vol. 975, no. 8887, str. 1-5, Mar. 2016.
- [12] 121212 H. Daumé III, "Unsupervised Learning for Natural Language Processing," u *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, 2015.

Popis tablica

Tablica 1. Rezultati različitih postavka CNN-a

Tablica 2. Rezultati različitih postavka VGG16

Tablica 3. Rezultati različitih postavka MobileNet-a

Popis slika

Slika 1. Gol

Slika 2. Sedmerac

Slika 3. Time-out

Slika 4. Pasivni napad

Slika 5. Isključenje

Slika 6. Raspodjela klasa

Slika 7. Dimenzije slika

Slika 8. Metrike evaluacije klasičnog CNN-a

Slika 9. Matrica konfuzije za klasični CNN

Slika 10. Metrike evaluacije VGG16 modela

Slika 11. Matrica konfuzije modela VGG16

Slika 12. Metrike evaluacije modela MobileNet

Slika 13. Matrica konfuzije modela MobileNet

Slika 14. Graf treniranja modela CNN

Slika 15. Metrike evaluacije modela CNN nakon optimizacije

Slika 16. Matrica konfuzije modela CNN nakon optimizacije

Slika 17. Graf treniranja modela VGG16

Slika 18. Metrike evaluacije modela VGG16 nakon optimizacije

Slika 19. Matrica konfuzije modela VGG16 nakon optimizacije

Slika 20. Graf treniranja modela MobileNet

Slika 21. Metrike evaluacije modela MobileNet nakon optimizacije

Slika 22. Matrica konfuzije nakon optimizacije modela MobileNet

Slika 23. Rezultati modela YOLO

Slika 24. Matrica konfuzije modela YOLO

Slika 25. „Batch“ za treniranje