

Препознавање на емоции во слика/видео

Ментор:

д-р Ивица Димитриовски

Изработил:

Леонид Трајкоски 211169

Содржина :

1. Апстракт
2. Вовед
3. Емоции
 - a. Опис и што претставуваат тие
 - b. Типови на емоции
 - c. Методи за детекција на емоции во слика
4. Алгоритми за детекција на емоции во слики
 - a. Convolutional Neural Networks (CNN)
 - b. Transfer learning
5. Алгоритам за детекција на лице во слика
6. Имплементација

1. Апстракт

Оваа семинарска работа има за цел да ја претстави имплементацијата на алгоритмите потребни за детекција на емоции во слика или видео преку демонстрација на соодветниот код. Во имплементацијата се користат повеќе библиотеки, имено: OpenCV, Tensorflow, Keras, Matplotlib. Сите овие библиотеки беа неопходни за обработката на слики/видео како и машинското учење потребно за имплементацијата на алгоритмите. Прво ќе биде даден вовед кои емоции ќе се препознаваат, ќе дефинираме како ќе ги препознаваме тие емоции и нивното значење и примена во компјутерската визија. Ќе бидат опишани алгоритмите и техники кои што ќе се користат во процесот на детекција на емоции.

2. Вовед

Емоциите се важен аспект во нашиот живот кои опфаќаат голем дел од него, со од таа причина нивното препознавање во дигиталниот свет станува се побитен и побитен. За таа цел со напредокот на технологијата гледаме голем развој во техниките и методите за препознавањето на емоциите во слики или видеа. Препознавањето на емоции од страна на компјутер е врзано со напредокот на вештачка интелигенција во 20-от век, но навистина почнува да се развива во 21-от век кога технологијата конечно ги достигнува техниките потребни за машинското учење, имено невронските мрежи и длабокото учење.

3. Емоции

а. Опис и што претставуваат тие

Емоции се психички процеси што го изразуваат човековиот однос кон предметите, настаните и другите луѓе, но и неговиот однос кон самиот себе и кон сопствените постапки и доживувања. Емоциите ја определуваат и насочуваат човековата активност. Постои разлика меѓу поимите емоција и чувство. Емоциите се јавуваат и кај луѓето и кај животните, а чувствата се емоции што се јавуваат само кај луѓето. Тие се резултат од човековиот социјален живот. Ние попрецизно ќе работиме со чувствата кај човекот и нивното предвидување. Во овој проект ние имаме определено 7 основни чувства одредени од Роберт Плутник врз кои ќе правиме предвидувања, и тоа: гнев, одвратност, страв, среќа, неутралност, тага и изненадување. Имено ќе се определуваме по дефиницијата за основните емоции.



Фигура 1 Основните типови на емоции

б. Типови на емоции

Како што претходно наведив ние ќе ги анализираме седумте основни емоции:

i.

Гнев: интензивна емоционална состојба која вклучува силен непријатен и некооперативен одговор на согледана провокација, повреда или закана. Лицето кое доживува лутина често ќе доживее физички ефекти, како што се зголемен пулс, покачен крвен притисок и зголемено ниво на адреналин и норадреналин. Изразите на лицето може да варираат од внатрешно свиткување на веѓите до целосно намуртено лице.

ii.

Одвратност е емоционален одговор на отфрлање или одбивност кон нешто потенцијално заразно или нешто што се смета за навредливо, невкусно или непријатно. Изразите на лицето вклучуваат спуштени веѓи, нос збрчкан, горната усна повлечена нагоре, опуштени усни.

iii.

Страв е интензивно непријатна емоција како одговор на согледување или препознавање на опасност или закана. Изразите на лицето често се појавуваат како раширени очи, кренати веѓи и малку отворена уста, лицето може да изгледа и напнато.

iv.

Среќа е позитивна емоција, која зрачи со топлина и задоволство. Кога некоја личност доживува среќа, тоа одразува чувство на благосостојба, радост и целокупно задоволство. Изразите на лицето кои што можат да се појавуваат се: мускулите околу очите затегнати, брчки околу очите, образите подигнати, аглите на усните подигнати дијагонално.

v.

Неутралност е израз на лицето кое се карактеризира со мирно и присебно однесување, лишено од било каков отворен емоционален акцент. Тоа одразува состојба на емоционална рамнотежа, каде што нема ниту изразено позитивно ниту негативно чувство.

vi.

Тага е емоционална болка поврзана или се карактеризира со чувства на неповолност, загуба, очај, тага, беспомошност, разочарување и тага. Изразите на лицето што се појавуваат се: Внатрешните агли на веѓите подигнати, очните капади опуштени, аглите на усните спуштени надолу.

vii.

Изненадување се јавува кога ќе најдеме на ненадејни и неочекувани звуци или движења. Изразите на лицето што се појавуваат се: веѓите кренати, очните капади подигнати, устата виси отворена, зениците проширени.

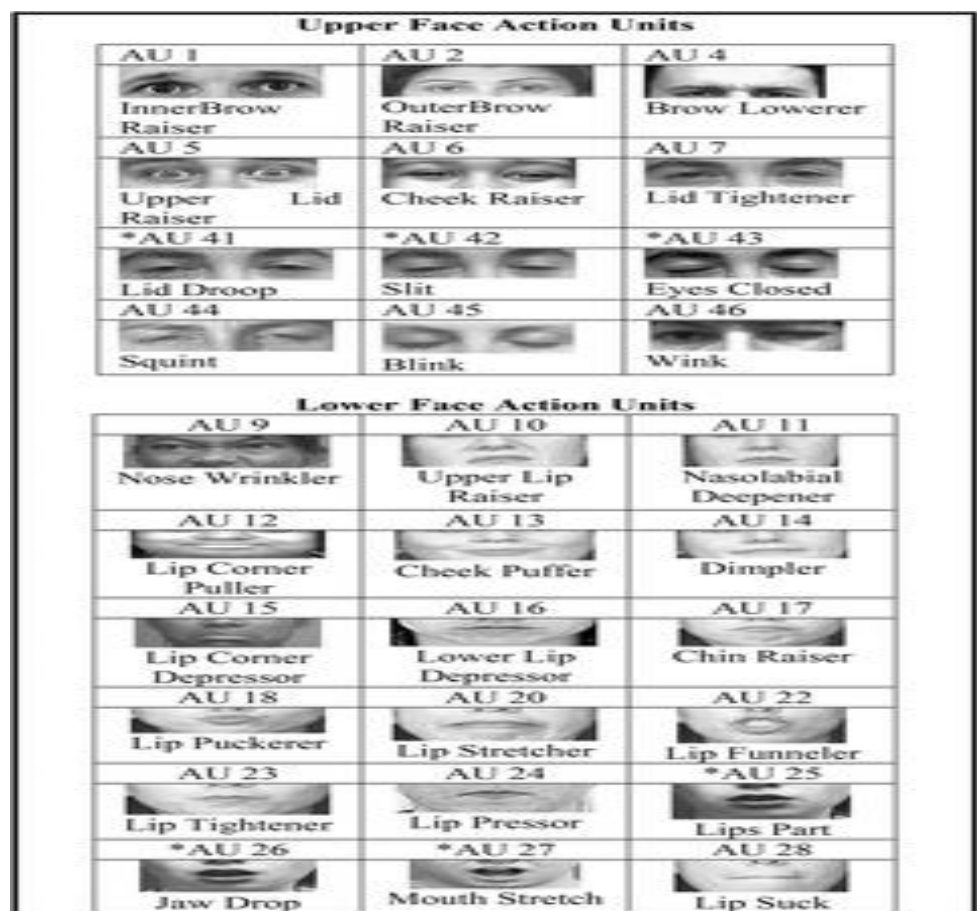
с. Методи за детекција на емоции во слика

i.

Препознавањето на изрази на лицето е задача за компјутерска визија која има за цел да ги идентификува и категоризира емоционалните изрази прикажани на човечко лице. Целта е да се автоматизира процесот на одредување на емоциите во реално време. Бидејќи препознавањето на изразите на лицето е потребно во секој алгоритам и метод за препознавање на емоции ќе обрне значително внимание. Користиме неколку методи за да успееме во ова задача вклучувајќи го **eigenspace** методот кој што е метод за намалување на димензионалноста на податоците, овој метод дава висока точност за откривање на лицето при променливи услови на поза.

Дополнително има и методи за препознавање на тенот на лицето наречен Adaptive skin colour method. Се користат неколку системи за боја вклучувајќи ги RGB, CMY, YIQ, YUV, YcbCr, но воглавно се користат YIQ и YUV, системи за боја. Во YIQ моделот за боја формулата за препознавање на тенот на кожа е $I = 0.596 \times R - 0.274 \times G - 0.322 \times B$. Поради големата и дополнителната процесирачка моќ потребна за препознавање на тенот на кожата ние ќе работиме со црно-бели слики, па овој метод нема да го користиме.

Други методи се Adaboost, Haar classifier, contours. Откако ќе се најде лицето, следно треба да се пронајдат изразите на лицето. Повторно користиме неколку методи, една од попознатите е методот на геометриска основа, тој се заснова на трајните карактеристики (очи, веѓи, чело, нос и уста) кои ја опишуваат формата и локацијата на компонентите на лицето користејќи однапред дефинирана геометриска позиција на обележје. Овие компоненти на лицето се извлекуваат за да се формира вектор на карактеристики што ја претставува геометријата на лицето. Следствено, основните изрази на лицето може да се идентификуваат со мерење на поместувањето на значајните компоненти на лицето.



Фигура 2 Метод на геометриска основа

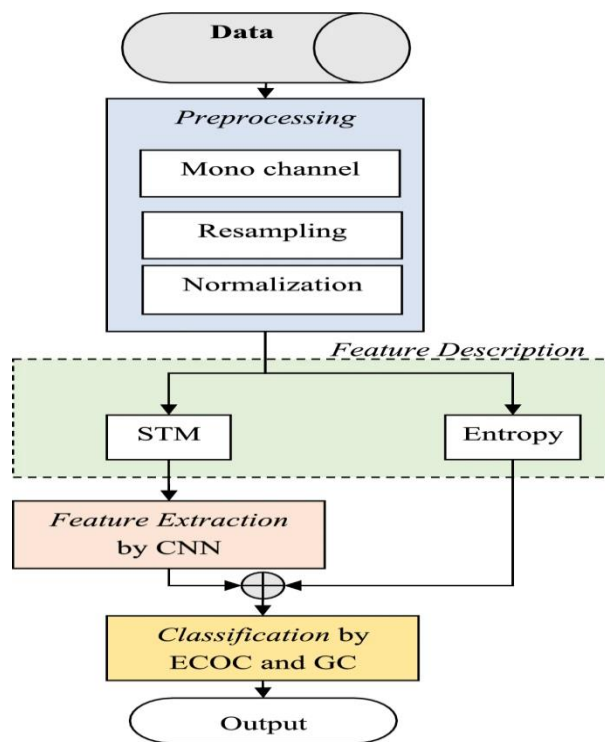
Друг метод е методот заснован на изглед, кој се фокусира на минливи карактеристики (брчки, испакнатини, преден дел) кои ги опишуваат промените во текстурата на лицето, интензитетот, хистограмите и вредностите на пикселите.

ii.

Гласовна анализа ја вклучува употреба на различни методи за анализа на вокалното однесување како маркер на афектот. Основната претпоставка на која се заснова оваа наука е дека има некои параметри кои што може да се идентификуваат и да се употребат за да се направи предвидување за емоционалната состојба на тој што зборува. Емоцијата во говорот може да се смета како комуникациски систем кој содржи неколку делови:

- 1.Изразување или прикажување на емоциите од говорникот (кодирањето)
- 2.Акустичните знаци (на пр., интензитетот на звукот) кои ја пренесуваат почувствуваната или планираната емоција
- 3.Проксималната перцепција на овие знаци од перципирачот (на пр., воочена гласност)
- 4.Заклучокот за изразената емоција од страна на перципирачот (декодирање).

Начинот на кој што се препознаваат емоциите е повторно со машинско учење, поточно невронски мрежи. Процесот започнува со тоа што откако ќе се претворат аналогните сигнали во дигитални податоците се препорецисраат, така што се претвораат во моно-канал, се средува фреквенцијата на сигналот, па потоа се нормализираат, следно се тренира невронската мрежа и на крај се предвидува.



Фигура 3 Класификација на емоции преку звук

iii. Машинско учење невронски мрежи и длабоко учење

Методите за препознавање на емоции во слики преку машинско учење и длабоко учење во моментот се најмоќните начини за препознавање на емоции. Техниките за машинско учење користат алгоритми за учење и генерализирање од етикетираните податоци, додека длабокото учење се фокусира на обука на вештачки невронски мрежи за автоматско учење на хиерархиски претстави. Методите за машинско учење често бараат рачно инженерство на карактеристики, додека методите за длабоко учење учат карактеристики директно од необработените податоци за пиксели. Моделите за длабоко учење, како што се Convolutional Neural Networks (CNN) како и учењето со трансфер се покажуваат како најдобри модели и алгоритми за препознавање на емоции во слика видео. Учењето со трансфер често се користи за да се користат претходно обучени модели. И двата методи бараат означени податоци за обука и пресметковни ресурси. Евалуацијата на моделот се врши за да се процени точноста и перформансите на обучените модели.

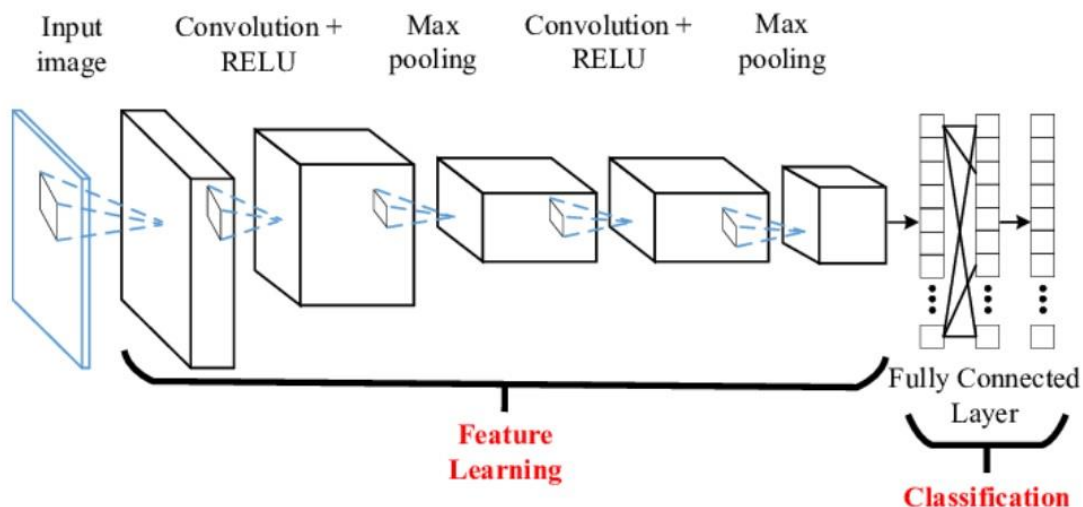
4. Алгоритми за детекција на емоции во слики

Во прилог се поставени неколку алгоритми и начини за детекција на емоции со користењето на машинско учење и длабоко учење, имено CNN и Transfer Learning.

a. Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) е технологија заснована на длабоко учење која има можности да постигне висока прецизност во препознавањето. Таа има повеќе слоеви, така што секој слој врши одредена функција на трансформација. “Convolutional” е првиот слој за извлекување карактеристики

од влезната слика. “Convolution” слојот потоа потоа ја зачувува врската помеѓу пикселите со функција за учење на слики користејќи мали квадрати од влезните податоци. “Convolution” на слика со различни филтри може да врши операции како што се откривање рабови, заматување и изострување со примена на филтри. Целта на ReLU е да се воведат нелинеарност во ConvNet. Вистинските податоци би сакале нашата ConvNet да ги научи да бидат ненегативни линеарни вредности. Следно, слојот за здружување функционира за да го намали бројот на параметри кога сликата е преголема. Просторното здружување исто така наречено subsampling или downsampling ја намалува димензионалноста на секоја мапа, но сепак задржува важни информации. Просторното здружување може да биде од различни типови како што е max pooling, average pooling или sum pooling. Целосно поврзаниот слој ја израмнува матрицата во вектор и ја внесува целосно поврзан слој како невронска мрежа. Слика 3 ја прикажува архитектурата на CNN.

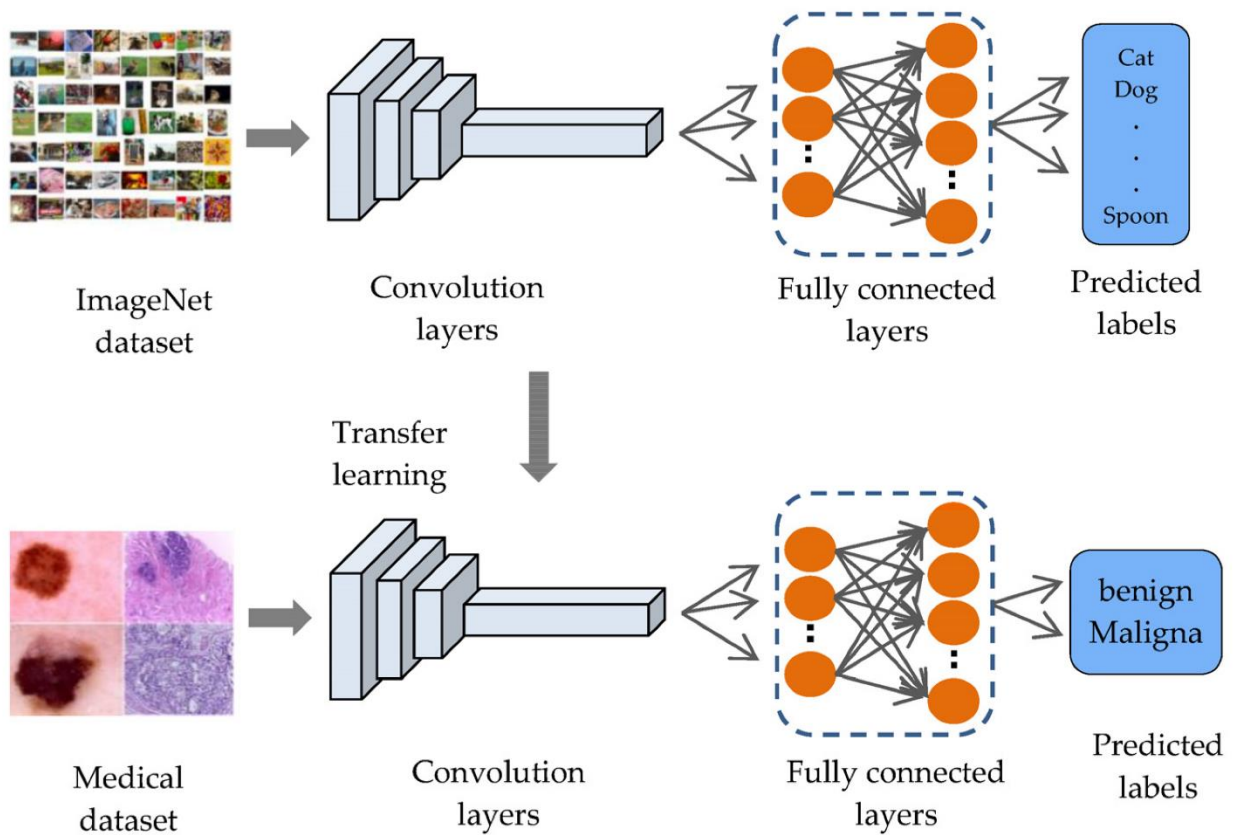


Фигура 4 Архитектурата на CNN слој

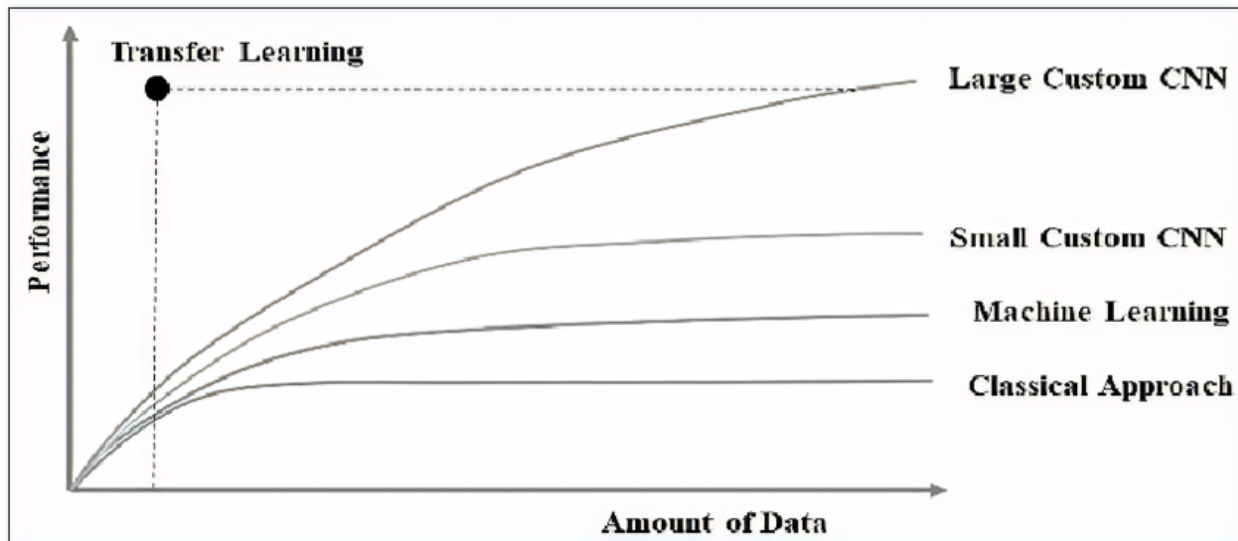
b. Transfer Learning

“Transfer Learning” е техника која го користи знаењето на обучен модел за да научи друг сет на податоци. “Transfer Learning” има за цел да го подобри учењето во целниот домен преку искористување на знаењето од изворниот домен и задачата за учење. Различни поставки за учење за пренос се дефинирани врз основа на видот на

задачата и природата на податоците достапни во изворниот и целниот домен. Поточно знаењето за веќе обучен модел на машинско учење се пренесува на различен, но тесно поврзан проблем во текот на учењето со пренос. На пример, ако сте обучиле едноставен класификатор да предвиди дали сликата содржи ранец, може да го искористите знаењето за обука на моделот за да идентификувате други предмети, како што се очилата за сонце.



Фигура 5 Преглед на Transfer learning



Фигура 6 Споредба на Transfer learning и CNN

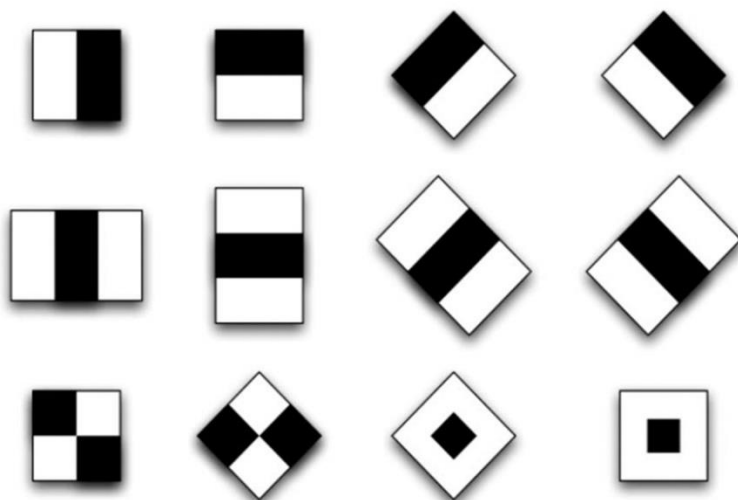
Очигледно е дека во општ случај Transfer learning дури и со мала количина на податоци ни дава далеку подобри резултати од CNN, само кога CNN е доволно голем и е доволно фино подесен може да се натпреварува и дава подобри резултати.

5. Алгоритам за детекција на фаца во слика

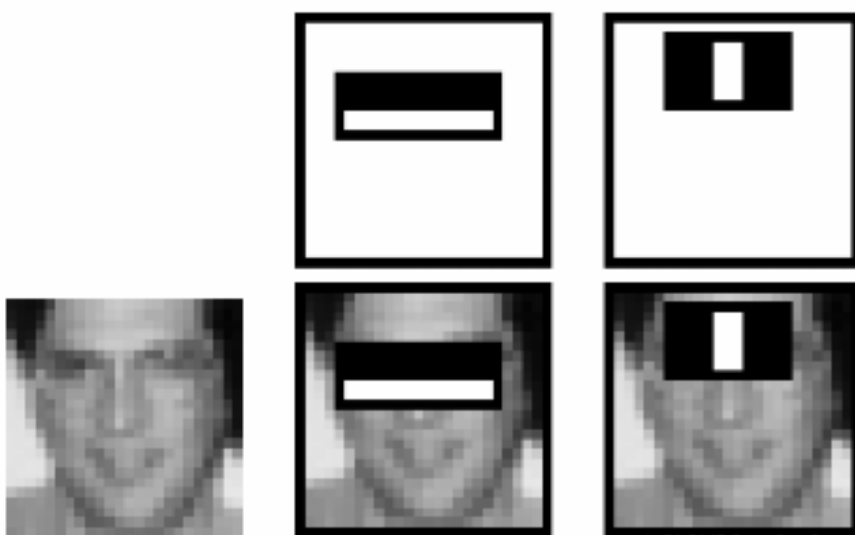
Ќе разгледуваме само еден алгоритам за детекција на лице и тоа е алгоритмот што е понуден од OpenCV библиотеката, имено Haar Cascade Classifier.

Идејата зад оваа техника вклучува користење на множество или каскада од класификатори за откривање на различни карактеристики на сликата. Овие класификатори потоа се комбинираат во еден силен класификатор кој може точно да разликува примероци што содржат човечко лице од оние што не содржат. Класификаторот Haar Cascade кој е вграден во OpenCV веќе е обучен на голема база на човечки лица.

Haar Classifier ги откриваат рабовите на лицето, линиите, движењата и бојата на кожата. Карактеристиките на Хаар се црно-бела поврзана правоаголна кутија како што се прикажани на следната слика



Фигура 7 Haar Classifier



Фигура 8 Примена на Haar Classifier

6. Имплементација

Во прилог на оваа семинарска ќе ги разгледаме алгоритмите методите и техниките користени за препознавањето на емоциите. Најпрво со средувањето на податочното множество, потоа ќе преминеме во тренирање на моделот, предвидување, следно преку користење на OpenCV библиотеката ќе препознаваме лица и ќе предвидуваме врз нив и на крај повторно преку користење на OpenCV библиотеката преку камера во живо ќе препознаваме емоции.

Сите бројки споменати надолу во текстот и сите тестирања спроведени се направени според познатото податочното множество на Kaggle -FER 2013 <https://www.kaggle.com/datasets/msambare/fer2013>.

Чекорите за имплементација на детекцијата на емоции се следни:

1. Средување на податочното множество
2. Користејќи tensorflow тренирање на моделот кој што може да биде CNN или Transfer learning
3. Имплементација на алгоритам за препознавање на лице од OpenCV
4. Имплементација на алгоритам за препознавање на лице на видео од OpenCV

Python имплементација

1. Пред да почнеме со тренирање на моделот прво мора да го спремиме податочното множество за обработка. Тоа го правиме така што прво сите слики од датотеките каде што се најдуваат ги вчитуваме и ги поставуваме во една низа

```
2. training_data=[]
3.
4. def create_training_Data():
5.     for category in Classes:
6.         path=os.path.join(Datadirectory,category)
7.         class_num=Classes.index(category)
8.         for img in os.listdir(path):
9.             try:
10.                 img_array=cv2.imread(os.path.join(path,img))
11.                 new_arrays=cv2.resize(img_array,(img_size,img_size))
12.                 training_data.append([new_array,class_num])
13.             except Exception as e:
14.                 pass
```

Поради големината на податочното множество и компјутерската моќ потребна, приморани бевме да го скратиме податочното множество за пола преку следниот код:

```
def delete_half_of_files_in_folders(parent_folder):
    for folder_name in range(7): # Assuming 7 folders from 0 to 6
        folder_path = os.path.join(parent_folder, str(folder_name))

        # Check if the folder exists
        if os.path.exists(folder_path):
            # Get a list of all files in the folder
            files = os.listdir(folder_path)

            # Calculate the number of files to delete (half of the total)
            num_files_to_delete = len(files) // 2
```

```

# Randomly select files to delete
files_to_delete = random.sample(files, num_files_to_delete)

# Iterate through selected files and delete them
for file_name in files_to_delete:
    file_path = os.path.join(folder_path, file_name)
    os.remove(file_path)
    print(f"Deleted: {file_path}")

# Specify the path to the parent folder containing folders 0 to 6
parent_folder_path = r'C:\Users\Leonid\Desktop\Seminarska\archive\train'

# Call the function to delete half of the files in each folder
delete_half_of_files_in_folders(parent_folder_path)

```

Следно треба да ја скалираме големината на податоците на 224, како и да ги нормализираме сите податоци .

```

X=[]
Y=[]
img_size=224
for features,label in training_data:
    X.append(features)
    Y.append(label)
X=np.array(X).reshape(-1,img_size,img_size,3)
#convert to 4d
X=X/255.0

```

Податоците се спремни за тренирање.

2.

Следно што треба да направиме е да го спремиме моделот за тренирање и да му додадеме број на на потребни слоеви, ние ќе користеме веќе трениран CNN, но со додавање на наши слоеви успеваме да имаме некој вид на Transfer learning

Input и Output слојот ги земаме директно од тренираниот модел.

```

base_input=model.layers[0].input
base_output=model.layers[-2].output

```

Следно поставуваме final output со додавање на наши слоеви.

```

final_output=layers.Dense(128)(base_output)
final_output=layers.Activation('relu')(final_output)#ACTVATION FUNC

```

```
final_output=layers.Dense(64)(final_output)
final_output=layers.Activation('relu')(final_output)
final_output=layers.Dense(7,activation='softmax')(final_output)#IMAME 7 FUNCKII
```

Откако ги додадовме слоевите потребни за работа следно почнуваме со тренирањето на моделот.

```
model=tf.keras.applications.MobileNetV2()
new_model=keras.Model(inputs=base_input,outputs=final_output)
new_model.compile(loss="sparse_categorical_crossentropy",optimizer="adam",metrics=['accuracy'])
new_model.fit(X,Y,epochs=5)
```

Користиме `sparse_categorical_crossentropy` функција на загуба, поради тоа што нашиот модел предвидува на повеќе категории не на две, додека користиме единствена метрика на прецизност која што се одредува како бројот на точни предвидувања врз вкупниот број на предвидувања.

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{All predictions}}$$

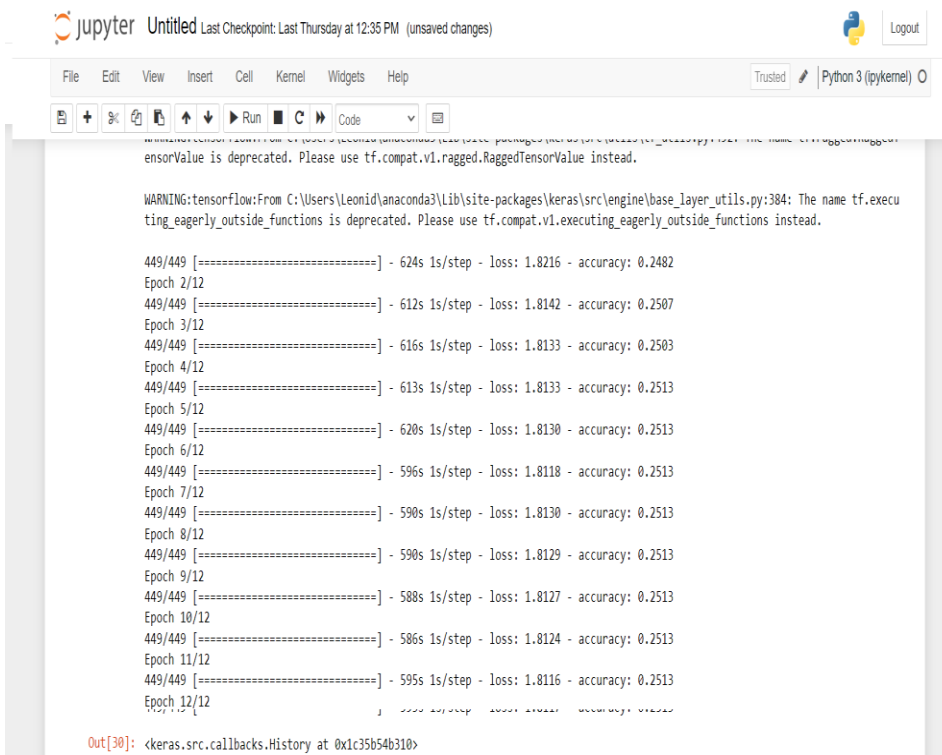


Повторно поради техничките ограничувања бевме приморани да стапиме до компромис така што намалениот број на податоци ќе доведе до не доволно трениран модел. Имено моделот достигнува локален максимум од кој што не може да излезе.

3.

Следно е алгоритмот за препознавање на лица. Најпрво вчитуваме слика врз која што ќе го применуваме алгоритмот, потоа го преку OpenCV библиотеката го повикуваме Haar Cascade методот за детекција на лице, потоа бидејќи нашиот модел е трениран на црно-бели слики ја претвораме сликата во црно бела и притоа проверуваме дали има повеќе лица во сликата, сите нив ги претвораме црно-бели и потоа на крај само лицето или фацата ја отсекуваме од сликата, ја зголемуваме да биде 224, 224 и ја нормализираме.

```
frame=cv2.imread("happy_boy.jpg")
face_cascade=cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
faces=face_cascade.detectMultiScale(gray,1.1,4)
for x,y,w,h in faces:
    roi_gray=gray[y:y+h,x:x+w]
    roi_color=frame[y:y+h,x:x+w]
    cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
    faces=face_cascade.detectMultiScale(roi_gray)
    if len(faces)==0:
        print("Face not detected")
    else:
        for(ex,ey,ew,eh) in faces:
            face_roi=roi_color[ey:ey+eh,ex:ex+ew]
final_image=cv2.resize(face_roi,(224,224))
final_image=np.expand_dims(final_image,axis=0)
```



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook is titled "Untitled" and shows the last checkpoint at 12:35 PM. The code cell contains the same Python script as shown in the previous block. The output cell displays training progress for 12 epochs, showing loss and accuracy values. The output is as follows:

```
Out[30]: <keras.src.callbacks.History at 0x1c35b54b310>
```

Epoch	Loss	Accuracy
1	1.8216	0.2482
2	1.8142	0.2507
3	1.8133	0.2503
4	1.8133	0.2513
5	1.8130	0.2513
6	1.8118	0.2513
7	1.8130	0.2513
8	1.8129	0.2513
9	1.8127	0.2513
10	1.8124	0.2513
11	1.8116	0.2513
12		


```
final_image=final_image/255.0
```

Конечно предвидуваме. Прво предвидуваме, потоа користејќи ја `argmax` функцијата од `numpy` ја земаме тежината со најголем стапка на предвидување и неа ја земаме како точно предвидување. Притоа задаваме вредности од 0-6 за која емоција претставуваат.

```
Predictions=new_model.predict(final_image)
```

```
predicted=np.argmax(Predictions)
```

```
if predicted==0:
```

```
    print("angry")
```

```
elif predicted==1:
```

```
    print("disgust")
```

```
elif predicted==2:
```

```
    print("fear")
```

```
elif predicted==3:
```

```
    print("happy")
```

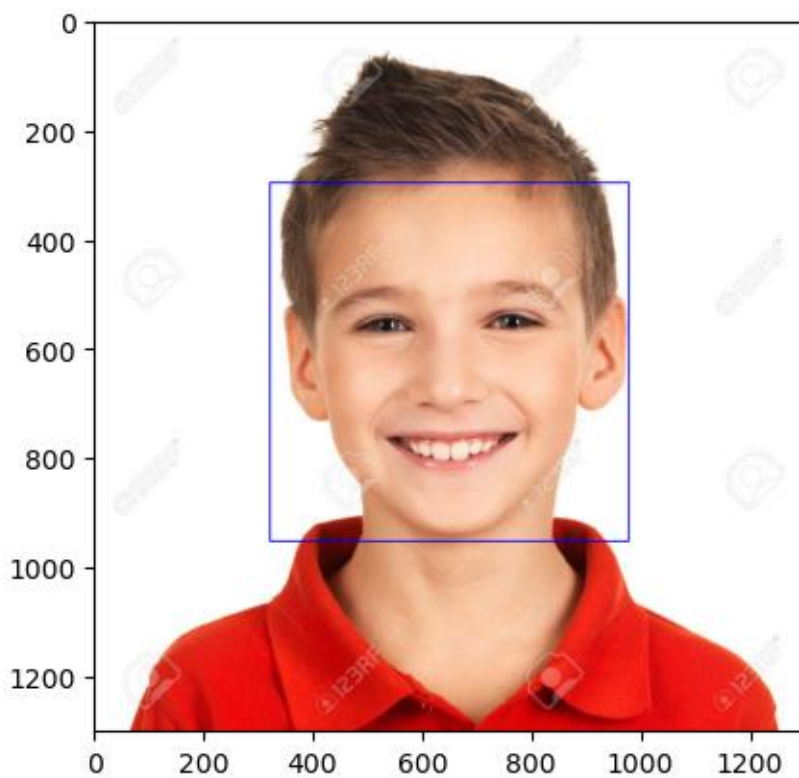
```
elif predicted==4:
```

```
    print("neutral")
```

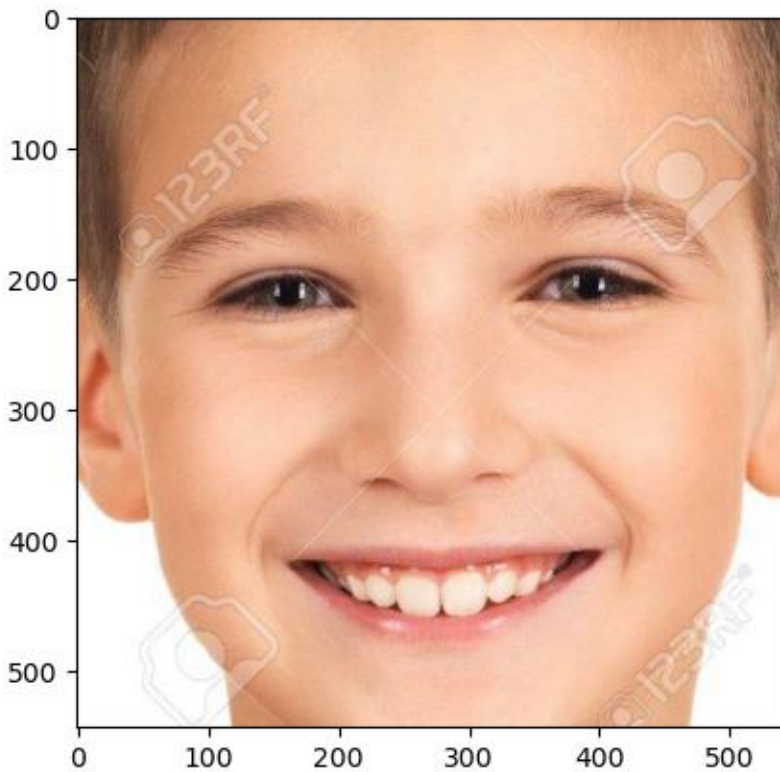
```
elif predicted==5:
```

```
    print("sad")
```

```
else: print("surprise")
```



Фигура 9 Препознавање на фаца



Фигура 10 Кастрирање на позадината

```
if predicted==0:
    print("angry")
elif predicted==1:
    print("disgust")
elif predicted==2:
    print("fear")
elif predicted==3:
    print("happy")
elif predicted==4:
    print("neutral")
elif predicted==5:
    print("sad")
else: print("suprise")
```

happy

Фигура 11 Успешно предвидување

Следно е вршењето на предвидувања во живо на камера. Кодот што се користи не се менува многу, освен во тоа што треба поставиме текст, да ја уклучиме камерата како и да предвидуваме во живо. Повторно користиме исклучиво OpenCV и тренираниот модел.

```

path='haarcascade_frontalface_default.xml'
font_scale=1.5
font=cv2.FONT_HERSHEY_PLAIN
rectangle_bgr=(255,255,255)
img=np.zeros((500,500))
text="Emotion"
(text_width,text_height)=cv2.getTextSize(text,font,fontScale=font_scale,thickness=1)[0]
text_offset_x=10
text_offset_y=img.shape[0]-25
box_coords=((text_offset_x,text_offset_y),(text_offset_x+text_width+2,text_offset_y-
text_height-2))
cv2.rectangle(img,box_coords[0],box_coords[1],rectangle_bgr,cv2.FILLED)
cv2.putText(img,text,(text_offset_x,text_offset_y),font,fontScale=font_scale,color=(0,0,0),thick
ness=1)

cap=cv2.VideoCapture(1)
if not cap.isOpened():
    cap=cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open webcam")

while True:
    ret,frame=cap.read()
    face_Cascade=cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default
.xml')
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces=face_Cascade.detectMultiScale(gray,1.1,4)
    for x,y,w,h in faces:
        roi_gray=gray[y:y+h,x:x+w]
        roi_color=frame[y:y+h,x:x+w]
        cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
        facess=face_Cascade.detectMultiScale(roi_gray)
        if len(facess)==0:
            print("Face not detected")
        else:
            for(ex,ey,ew,eh) in facess:
                face_roi=roi_color[ey:ey+eh,ex:ex+ew]
            final_image=cv2.resize(face_roi,(224,224))
            final_image=np.expand_dims(final_image,axis=0)
            final_image=final_image/255.0
            font=cv2.FONT_HERSHEY_SIMPLEX
            Predictions=new_model.predict(final_image)
            font_scale=1.5
            font=cv2.FONT_HERSHEY_PLAIN
            predict=np.argmax(Predictions)

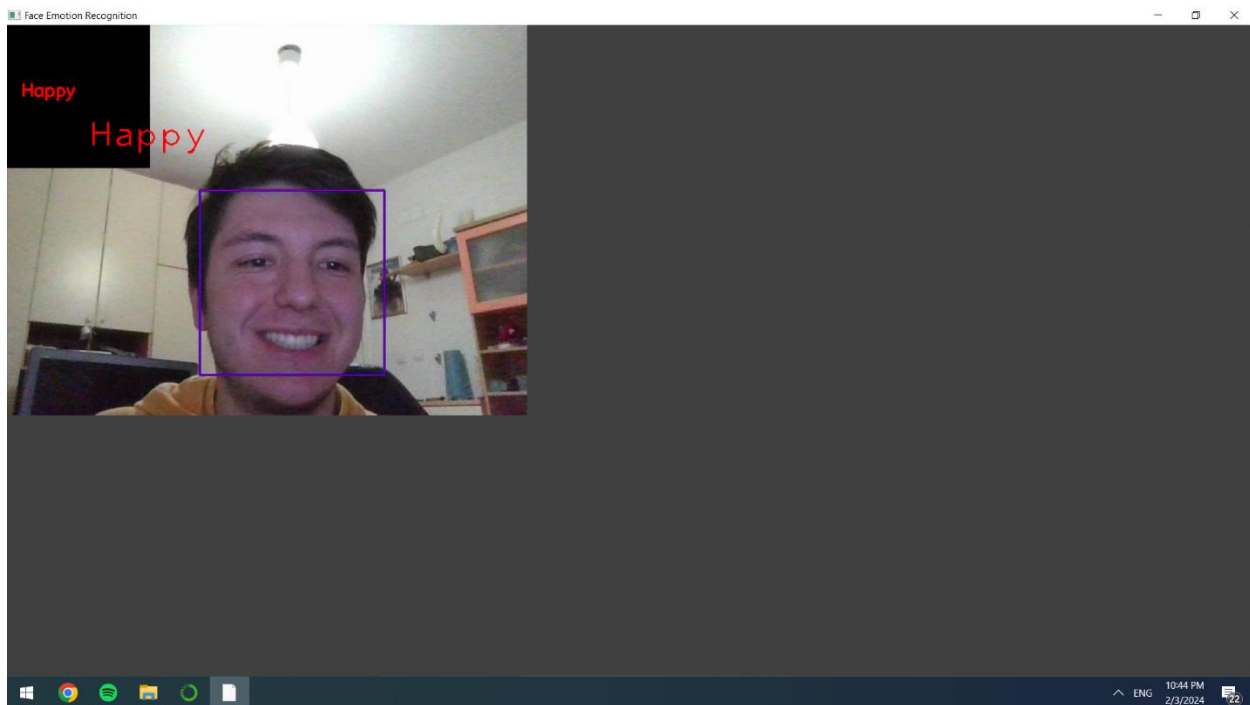
```

```

elif predict==3:
    status="Happy"
cv2.imshow("Face Emotion Recognition",frame)
if(cv2.waitKey(2)& 0xFF==ord('q')):
    break
cap.release()
cv2.destroyAllWindows()

```

Исто и како претходно им задаваме вредност на категориите од 0-6, но со цел да се минимализира кодот се друго е прескокнато.



Фигура 12 Препознавање на емоции во живо

Заклучок

Во рамките на оваа семинарска ги разгледавме основните емоции, накратко ги објаснивме потоа разгледавме како да ги детектираме и пронајдеме. Препознавањето на емоции е актуелна тема која што има примени од заштита до подобрувањето на реклами до медицината. Разгледавме неколку начини за препознавањето на емоциите и стигнавме до заклучокот дека се со цел да успееме мораше да донесиме некој компромис.

Преку оваа семинарска работа се покажа еден едноставен начин како да имплементираме еден алгоритам за детекција на емоции во слика/видео. Исто така, беше разгледана имплементација на веќе готови алгоритми со помош на библиотеки. Изборот на соодветни алгоритми и модели останува релативна тема и се донесува според потребите.

Референци

- [1] <https://www.sciencedirect.com/science/article/abs/pii/S2214785321048987>
- [2] <https://github.com/omarsayed7/Deep-Emotion>
- [3] <https://www.youtube.com/watch?v=yN7qfBhfGqs>
- [4] <https://www.paulekman.com/universal-emotions/what-is-surprise/>
- [5] <https://paperswithcode.com/task/facial-expression-recognition>
- [6] <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-ipr.2018.6647>
- [7] <https://www.learndatasci.com/glossary/eigenspace/>
- [8] http://www.scholarpedia.org/article/Speech_emotion_analysis
- [9] <https://www.nature.com/articles/s41598-023-47118-4>
- [10] Wan, Jing & Yuan, ze & Zhang, Xuechao. (2015). Adaptive Skin Color Detection Based on Human Face under Complex Background. MATEC Web of Conferences. 22. 01006. 10.1051/mateconf/20152201006.
- [11] <https://www.mdpi.com/2076-3417/13/17/9890>
- [12] Kanagaraju, P. & Ranjith, M. & Vijayasarathy, K.. (2022). Emotion detection from facial expression using image processing. International journal of health sciences. 1368-1379. 10.53730/ijhs.v6nS6.9748.
- [13] <https://iopscience.iop.org/article/10.1088/1742-6596/1962/1/012040>
- [14] <https://www.sciencedirect.com/topics/computer-science/transfer-learning>