

**ДЕТЕКЦИЈА ЕМОЦИЈА ОПТИЧКИМ ПУТЕМ АНАЛИЗОМ
ФАЦИЈАЛНИХ ЕКСПРЕСИЈА
OPTICAL METHOD FOR EMOTION DETECTION BY ANALYZING FACIAL
EXPRESSIONS**

Немања Томовић, др Владимир Рајс, Факултет техничких наука, Нови Сад

Област – ЕЛЕКТРОНИКА И РАЧУНАРСТВО

Кратак садржај – У овом раду приказана су психолошка знања и истраживања из области људских емоција, при чему се ове информације користе како би се пронашло техничко решење проблема детекције емоција на основу фацијалне експресије. Рад обрађује више метода, њихове предности и недостатке; имплементирајући методу која помоћу камере и алгорита машинског учења перципира емоције.

Кључне речи: емоције, физички и физиолошки процеси, сензори, машинско учење, неуронске мреже, *raspberry pi, python, tensorflow, keras*

Abstract – *In this paper knowledge and researches gained from the field of psychology of human emotion are shown alongside the important problem solving implementation-necessary knowledge. The paper processes multiple methods, as well as the advantages and disadvantages; implementing the method which provides solution to the problem using camera module and machine learning algorithm.*

Keywords: *emotion, physical and physiological processes, sensors, machine learning, neural networks, raspberry pi, python, tensorflow, keras*

1. УВОД

Емоције су компликовани процес који се одиграва у телу појединца. У данашње време наука добро познаје основне принципе настанка емоција, али недовољно добро да би их описала математичким алатом [1]. Сходно томе, алгоритам машинског учења је идеалан за решавање поменутог типа проблема. Неопходно је изазвати појаву емоције код испитаника, у контролисаним условима и сачувати те информације које ће касније бити кључне за обуку неуронских мрежа. У овом раду истражени су принципи детекције биосигнала и телесних експресија услед промене емотивног стања. Фокус имплементације је постављен на детекцији телесних промена (фацијалних промена, тзв. фацијалних експресија) због своје једноставности, исплативости и практичности. У наставку биће приказана и имплементација једног таквог алгорита.

НАПОМЕНА:

Овај рад проистекао је из мастер рада чији ментор је био др Владимир Рајс, ванр. проф.

1.1. Биосигнали

Биосигнали су свеприсутни у људском телу и представљају „одговор“ рада одређеног органа. На пример, срце генерише импулсе које називамо откуцаји срца, док мозак генерише магнетно поље услед електричних активности које се у њему догађају [2]. Поремећај у раду биосигнала је приметан услед промене емотивног стања, те се они могу користити за детекцију емоције. Ипак, метода није практична јер захтева велики број сензора, габаритних је димензија и крута.

1.2. Телесне експресије

Покрети могу бити вољни и безвољни. Вољни покрет је покрет који правимо свесно, на пример писање. Безвољни покрет не можемо свесно да контролишемо и последица је реакције аутономног нервног система. При промени емотивног стања, органи почињу да раде абнормално услед чега се лучи много већи број хормона што се коначно испољава кроз безвољну телесну реакцију (поскакивање, смејање, подрхтавање и слично). Перципирањем телесних експресија може се донети закључак коју емоцију осећа појединац [3]. Недостатак ове методе је што манипулативна понашања, где појединац свесно имитира телесне експресије зарад исказивања емоције коју не осећа, ће преварити систем.

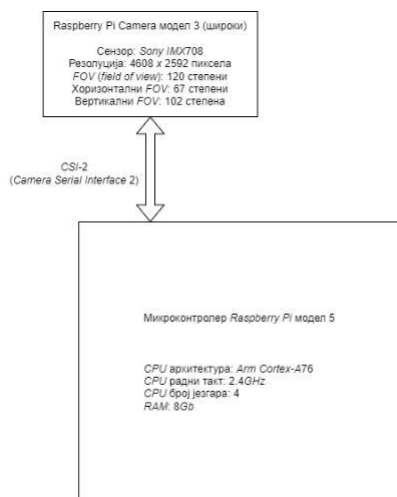
1.3. Перцепција и машинско учење

Временом сваки појединац научи да перципира ствари око себе. Једна од тих ствари су људске емоције. Перцепција се одвија несвесно што се доказује тиме да човек иако може да да тачан одговор, тешко му је да објасни зашто је баш тај одговор дао. Иста ситуација је и код машинског учења. Машина се дају проверени подаци; за које се зна шта представљају. Са порастом примера, машина временом почиње да перципира податке и исправно препознаје шта се на подацима налази. Покушајем анализирања одговора неуронске мреже, која имплементира алгоритам машинског учења, наилази се на зид. Исто као човек, машина није свесна зашто је дала баш тај одговор, али сигурна је у њега. Ипак, постоји мала разлика која разграничава човека од машине. Машинско учење је исказ математичког језика, комплексна функција, те машина увек квантификује сигурност свог одговора (0-100%), док човек, у најбољем случају, то може да учини само описно.

2. ПРАКТИЧНА РЕАЛИЗАЦИЈА

2.1. Реализација на микропроцесорској јединици

На слици 1 приказана је практична реализација проблема коришћењем методе анализе телесних експресија уз ограничење – коришћен је само подскуп информација (фацијална експресија). Овај подскуп даје највише потребних информација за решавање проблема детекције емоционалног стања. Остатак информација, понаособ гледано, нису довољни за решавање проблема, али могу пружити већу сигурност у тачност одговора.



Слика 1: Блок шема реализације детекције људских емоција анализом фацијалних експресија

Да би се проблем решио користи се оптички сензор (камера) који прибавља слику и предаје је микропроцесорској јединици (raspberrypi) на даљу обраду. На микроконтролеру налази се обучена неуронска мрежа за решавање проблема класификације емоција. Улазна слика се обрађује и предаје неуронској мрежи. Улазна слика у неуронску мрежу је величине 224x224 пиксела у троканалном RGB (енгл. red green blue, срп. црвена зелена плава) формату.

```
41 emotion_recognition_model = tf.keras.models.load_model ("neural_network/v8.h5")
42 face_cascade = cv2.CascadeClassifier (
43     cv2.data.haarcascades +
44     "haarcascade/haarcascade_frontalface_default.xml"
45 )
46
47 cv2.namedWindow ("Full Screen", cv2.WND_PROP_FULLSCREEN)
48 cv2.setWindowProperty ("Full Screen", cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
49
50
51 picamera = Picamera ()
52 picamera.configure (picamera.create_preview_configuration ( main = {
53     "format": "RGB888",
54     "size": (1024, 600),
55     "$size": (4608, 2592)
56 })
57 )
58
59 picamera.start ()
60
```

Код 1: Конфигурација и креирање потребних објеката за програм за детекцију људских емоција

Са кода 1 приметно је учитавање модела неуронске мреже за класификацију емоција (линија 42), модела за детекцију људског лица, базираног на Харовој таласној трансформацији (линија 43), као и конфигурација и започињање рада камера модула (линије 52, 59).

```
61 while True:
62     image_RGB = picamera.capture_array ()
63     image_grayscale = cv2.cvtColor (image_RGB, cv2.COLOR_RGB2GRAY)
64     detected_faces = face_cascade.detectMultiScale (
65         image_grayscale,
66         HAARCA cascade_SCALE_FACTOR,
67         HAARCA cascade_MIN_NEIGHBOURS,
68         minSize = (50, 50)
69     )
70
71     roi = find_face_roi (detected_faces, image_RGB, image_grayscale)
72     if (roi is not None):
73         image_nn = cv2.resize (roi, (IMAGE_SIZE, IMAGE_SIZE))
74         #image_nn = image_nn.reshape (1, IMAGE_SIZE, IMAGE_SIZE, 3)
75         image_nn = np.expand_dims (image_nn, axis = 0)
76         image_nn = image_nn / 255.0
77
78         predictions = emotion_recognition_model.predict (image_nn)
79         predicted = np.argmax (predictions)
```

Код 2: Runtime програма – преузимање улазних информација и обрада, проналазак лица на слици и класификација емоције путем неуронске мреже

Са кода 2 види се да микропроцесорска јединица (raspberrypi) преузима слику са камере (линија 62). Ова слика се потом обрађује у једноканални монохроматски формат (линија 63) како би се прилагодила облику потребном да Харова таласна трансформација пронађе локацију лица на слици (линија 64), уколико оно постоји. Лице се изолује од остатка слике (линија 71), величина слике се прилагођава (линија 73) и формат информација прилагођава улазу неуронске мреже за класификацију емоција (линије 75, 76) након чега се шаље упит овој мрежи (линија 78) и анализира њен одговор (линија 79).

```
81
82 match (predicted):
83     case 0:
84         emotion = "bes"
85     case 1:
86         emotion = "gadjenje"
87     case 2:
88         emotion = "strah"
89     case 3:
90         emotion = "sreca"
91     case 4:
92         emotion = "neutralno"
93     case 5:
94         emotion = "tuga"
95     case 6:
96         emotion = "iznenadjenje"
97     case _:
98         emotion = "unknown"
```

Код 3: Runtime програма – мапирање одговора неуронске мреже на људски разумљив језик

Код 3 приказује мапирање одговора неуронске мреже за класификацију емоција у одговор смислен човеку. Наиме, мрежа даје одговор у систему бројевних вредности од 0 до 6 што се даље мапира у људски разумљиве информације:

- бес (0);
- гађење (1);
- страх (2);
- срећа (3);
- неутрално (4);
- туга (5);
- изненађење (6).

Мапирање емоција на број није произвољно изведено. Оно зависи од начина на који је мрежа обучена и шта јој је речено да свака од ових бројних вредности представља. У наредном поглављу ће бити приказана обука неуронске мреже.

2.2. Реализација обуке неуронске мреже

На коду 4 може се видети мапирање означених бројевних вредности излаза неуронске мреже на литералну вредност емоционалне класе коју представљају.

Basic Information for Neural Network

```
[25]: training_dataset_location = "train"

[26]: testing_dataset_location = "test"

[27]: prediction_classes = ["angry", "disgust", "fear", "happy", "neutral", "sad", "surprise"]
# 0 - angry
# 1 - disgust
# 2 - fear
# 3 - happy
# 4 - neutral
# 5 - sad
# 6 - surprise
```

Код 4: Мапирање излаза на људски читљиву вредност

Приликом обуке неуронске мреже кориштен је алат *jupyter notebook*. Алат се показао као идеално решење због недовољно *RAM* (енгл. *Random Access Memory*) ресурса. Инструкције су често извршаване у фазама како би се омогућило програму да се успешно изврши. Алгоритми машинског учења су поприлично захтевни за рачунар захтевајући велики меморијски утрошак зарад учитавања података за обуку мреже и математичког прорачуна параметара мреже који се обучавају. Код 5 приказује учитавање целокупног сета за обуку неуронске мреже. Напомиње се да сет за валидацију неуронске мреже није кориштен у овом раду због недостатка меморијског простора.

Load Training Dataset

```
[12]: for recognition_element in prediction_classes:
dataset_path = os.path.join(training_dataset_location, recognition_element)
emotion = prediction_classes.index(recognition_element)

for image_of_emotion in os.listdir(dataset_path):
    try:
        image_data = cv2.imread(os.path.join(dataset_path, image_of_emotion))
        image_data = cv2.resize(image_data, (IMAGE_SIZE, IMAGE_SIZE))
        training_data.append((image_data, emotion))
    except Exception as e:
        pass

[13]: random.shuffle(training_data)
```

Код 5: Учитавање сета за обуку неуронске мреже

Код 6 приказује екстракцију података из сета за обуку. Подаци из овог сета се раздвајају на слику, која приказује одређену емоцију (*features* објекат), и ознаку, о којој емоцији је реч (*labels* објекат). Са истог кода може се приметити да је у току имплементације праћен утрошак меморије и ресурси су експлицитно ослобађани.

Extract Features and Labels from Training Dataset

```
[17]: for f, i in training_data:
features.append(f)
labels.append(i)

features = np.array(features).reshape((NP_ARRAY_RESHAPE_SHAPE_CALCULATED_TUPPLE, IMAGE_SIZE, IMAGE_SIZE, NP_ARRAY_RESHAPE_DIMENSION))
labels = np.array(labels)

[18]: (check_memory_consumption())
Changed RAM consumption: 3.042089572753066 GB

Approximately additional 4GB of RAM is allocated for the application after extracting training_data into features and labels arrays.
Total consumption of application is 4GB.
In order to free some space, training_data will be immediately removed (not waiting for garbage collector) as it will not be used in future references since it was extracted to features and labels array.

[19]: del training_data

[20]: (check_memory_consumption())
Changed RAM consumption: 4.235793980224689 GB

after training_data has been removed we can observe that 4GB was freed from the application leading to the fact that application requires approximately 4GB to run.
```

Код 6: Екстракција сета за обуку у објекат података за обуку и објекат ознака класе за класификацију

Код 7 приказује прилагоду улазног податка. Прва инструкција означена као [50], је извршена како би се смањила алокација меморије, опет.

Типично, вредности из реалног система се смештају у максималну ширину меморије одређену архитектуром рачунара (у случају коришћеног лаптопа *Victus* ради се о 64-битним меморијским ћелијама). Накнадна инструкција представља нормализацију податка из [0, 255] (осмобитни *RGB* формат) у [0, 1] (скуп реалних бројева са којима ради мрежа).

Код 7: Прилагода и нормализација података типа слика

При обуци кориштен је принцип преносног учења. Изабран је модел *MobileNetV2*, који омогућава класификацију до 1000 различитих класа слика. Овај модел је обучен над подацима типа слике, због чега се за било који проблем тог типа може искористити зарад убрзавања процеса учења. Учитавање је приказано кодом 8. Код 9 приказује прилагоду излаза са 1000 класа на 7 (онолико колико се у овом раду детектује). Додатно, излазни слој није замењен једним од 7 неурона, већ са три слоја са по 128, 64 и 7 неурона (респективно од скривеног слоја ка излазу гледано). Број замењених слојева и број изабраних неурона је произвољно одабран.

Deep Learning Using Transfer Learning

```
[7]: pretrained_model = tf.keras.applications.MobileNetV2()
```

Код 8: Учитавање претренираног модела за решавање проблема класификације слика *MobileNetV2*

```
[24]: # adding new layer after the output of global pooling layer
new_output_layer = layers.Dense(128, activation = 'relu')(pretrained_model.output)

[25]: # adding new layer after the output of global pooling layer
new_output_layer = layers.Dense(64, activation = 'relu')(new_output_layer)

[26]: # 7 classes to be classified in total
new_output_layer = layers.Dense(7, activation = 'softmax')(new_output_layer)
```

Код 9: Прилагода излазног слоја учитане мреже како би одговарала проблему класификације 7 класа

Извршене промене су, потом, учитане, што приказује код 10, чиме је формирана неуронска мрежа за решавање проблема класификације емоција.

Create New Model

```
[37]: emotion_recognition_model = keras.Model(inputs = pretrained_model.input, outputs = new_output_layer)
```

Код 10: Учитавање измена у модел неуронске мреже

Код 11 приказује један од корака обуке неуронске мреже. Обука је вршена у фазама. Укупно је извршено 30 фаза у којима се узима 1/32 укупног броја података за обуку. Накнадно је обављено још 7 фаза где су сви подаци из мреже за обуку дати одједном.

Step 37

```
[10]: emotion_recognition_model_v2 = tf.keras.models.load_model("v16.h5")
emotion_recognition_model_v2.compile(loss = "sparse_categorical_crossentropy", optimizer = "adam", metrics = ["accuracy"])
emotion_recognition_model_v2.fit(features, labels, batch_size = 1, epochs = 15)
emotion_recognition_model_v2.save("v17.h5")

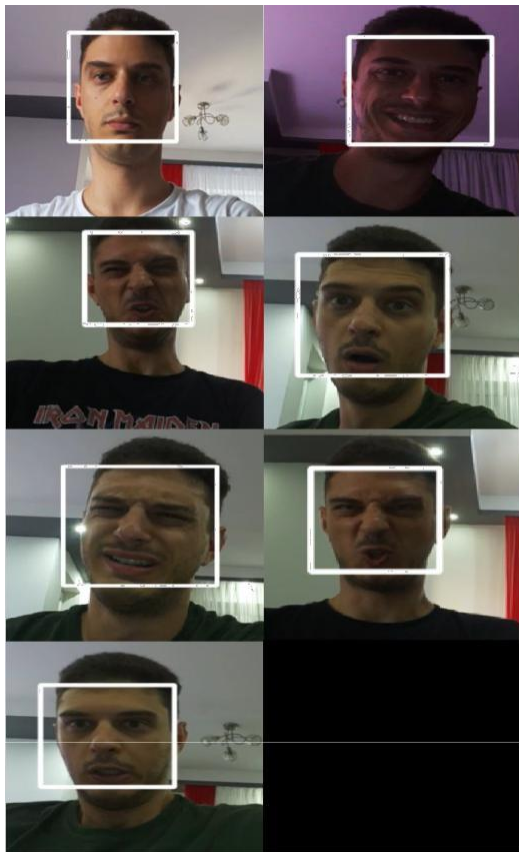
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. 'model.compile_metrics' will be empty until you train or evaluate the model.

Epoch 1/15
28/280/28000 --- 28384 70ms/step --- accuracy: 0.9214 - loss: 0.2705
Epoch 2/15
28/280/28000 --- 33734 117ms/step --- accuracy: 0.9101 - loss: 0.2844
Epoch 3/15
28/280/28000 --- 46364 115ms/step --- accuracy: 0.9104 - loss: 0.2880
Epoch 4/15
28/280/28000 --- 44474 115ms/step --- accuracy: 0.9104 - loss: 0.2792
```

Код 11: Приказ једне фазе обуке неуронске мреже

3. РЕЗУЛТАТИ

У овом одељку дат је приказ неких од добијених резултата имплементираних решења. На слици 2 приказана је по једна слика сваке емоције добијена у различитим оперативним условима. У првој колони се могу видети неутрално стање, гађење, туга и страх, док се у другој колони могу видети срећа, изненађење и бес.



Слика 2: Приказ резултата имплементираних мрежа над свим класама

4. ЗАКЉУЧАК

Детекција емоција оптичким путем је изузетно практична метода јер не захтева посебно дизајниране круте уређаје габаритних димензија. Довољно је поставити камеру и пустити алгоритам да одради остатак посла. Практична примена ове апликације се изналази само у информативним и забавним сврхама. Унапређење апликације се може одвити у два смера. У првом смеру апликација би се побољшала додавањем информација о говору тела (подскуп информација који је изузет приликом реализације пројекта), чиме би се могле открити неконзистентности између претварања и стварног осећања одређене емоције, стим да се не решава проблем манипулативног понашања. Други смер унапређења захтева увођење бесконтактних сензора којима се могу детектовати биосигнали (на пример: термалном камером се може детектовати промена телесне температуре која је праћена емоционалном променом). На овај начин систем може осигурати детекцију манипулативног понашања.

Иако је у раду коришћено шест емоција, које представљају базичне емоције по Екману, постоји и треће унапређење које се може извести. Екман и други психолози дефинишу емоције у две групе: базичне и комплексне [4]. Рад се додатно може проширити коришћењем *Microsoft*-ове базе података која убраја чак 135 класа емоција на основу фацијалних експресија.

4. ЛИТЕРАТУРА

- [1] Don H. Hockenbury, Sandra E. Hockenbury, "Discovering psychology", *Worth Publishers*, април 2006.
- [2] Steven J. Haggblom, Renee Warnick, Jason E. Warnick, Vinessa K. Jones, Gary L. Yarbrough, Tenea M. Russell, Chris M. Borecky, Reagan McGahhey, John L. Powell III, Jamie Beavers, Emmanuelle Monte, "Review of General Psychology", *American Psychology Association*, јун 2002.
- [3] Manuela Angioni, Franco Tuveri, "Discovering Emotions through the Building of Linguistic Resource", *Center for Research and Scientific Studies in Sardinia, Italy*, 2019.
- [4] Robert Plutchnik, Henry Kellerman, "Theories of Emotion volume 1", *Academic Press*, 1980.

Кратка биографија

Немања Томовић рођен је априла 1996. године у Новом Саду у Републици Србији. Одрастао је у Зрењанину, где је прве програмерске кораке начинио са једанаест година похађајући приватну школу за рачунарство и технику, а у Нови Сад се враћа због студирања. Завршио је смер мехатроника на основним академским студијама Факултета Техничких Наука дипломирајући из области компјутерске визије за детекцију коловозних линија са циљем аутоматске вожње. Годинама, током студија, али и након завршетка, бавио се израдом робота за Eurobot такмичење и био један од оснивача непрофитабилног удружења Invictus Robotics Association. Тренутно је запослен као софтверски архитекта за сигурност и безбедност возила.

Владимир Рајс рођен је 1982. године у Апатину. Дипломирао је 2007, а докторирао 2015. године на Факултету техничких наука у Новом Саду. Од 2016. године је био запослен као доцент, од 2021. као ванредни професор на Департману за електронику, енергетику и телекомуникације ФТН-а.

UDK: 621.38

DOI: <https://doi.org/10.24867/31BE04Tomovic>