

Korisnost pretvorbe vremenskih podataka u sliku u svrhu klasifikacije

Katić, Josip

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:166:286875>

Rights / Prava: [Attribution-NonCommercial 4.0 International/Imenovanje-Nekomercijalno 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-03-10**

Repository / Repozitorij:

[Repository of Faculty of Science](#)



SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**Korisnost pretvorbe vremenskih podataka u sliku u
svrhu klasifikacije**

Josip Katić

Split, rujan 2022.

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu

Prirodoslovno-Matematički fakultet

Odjel za informatiku

Ruđera Boškovića 33, 21000 Split, Hrvatska

Korisnost pretvorbe vremenskih podataka u sliku u svrhu klasifikacije

Josip Katić

SAŽETAK

Duboko učenje je jedno od područja računalstva koje je postalo sve popularnije i razvijenije posljednjih godina i to je područje koje se promatra unutar ovog rada. Točni aspekti koji se promatraju je usporedba dvije arhitekture neuronskih mreža. Promatramo rekurentnu i konvolucijsku neuronsku mrežu (RNN i CNN) s naglaskom na konvolucijsku neuronsku mrežu u smislu da koristimo podatke koji nisu originalno namijenjeni za tu arhitekturu te hoće li ta mreža s prilagodbom podataka postići zadovoljavajuće rezultate koji su usporedivi s arhitekturom i podacima korištenim za rekurentnu neuronsku mrežu. Korišteni podatci su snimke moždanih valova dobivene koristeći EEG namijenjeni za rad s RNN arhitekturom, te se prilagodba podatka za CNN arhitekturu izvodila koristeći GAF (GASF i GADF) i MTF algoritme za generiranje slika iz vremenskih podataka.

Ključne riječi: RNN, CNN, duboko učenje, EEG, GAF, GASF, GADF, MTF

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 44 stranice, 29 grafičkih prikaza, 7 tablica i 14 literaturnih navoda.

Izvornik je na hrvatskom jeziku

Mentor:

Dr.sc. Saša Mladenović, *izvanredni profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Ocjenjivači:

Dr.sc. Saša Mladenović, *izvanredni profesor Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Dr.sc. Divan Krpan, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Dr.sc. Goran Zaharija, *docent Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu*

Rad prihvaćen: Rujan 2022

Basic documentation card

Thesis

University of Split

Faculty of Science

Department of Computer science

Ruđera Boškovića 33, 21000 Split, Croatia

Utility of converting time series data into an image for classification purposes

Josip Katić

ABSTRACT

Deep learning is one of the fields of computer science that has become ever more popular and advanced in recent years and that is the field which is observed within this paper. Exact aspect that is observed is the comparison of two neural network architectures. We observe the recurrent and the convolutional neural network (RNN and CNN) while concentrating on the convolutional neural network in the sense that we use data that is not originally suitable for that architecture and will that network with the adjusted data give a satisfying results that are comparable with the architecture and data used for a recurrent neural network. The used data are recordings of brain waves that are acquired by using an EEG and are intended for RNN architecture but for the CNN architecture the data was adjusted by using GAF (GASF and GADF) and MTF algorithms for generating images from time series data.

Key words: RNN, CNN, deep learning, EEG, GAF, GASF, GADF, MTF

Thesis deposited in library of Faculty of science, University of Split

Thesis consists of: 44 pages, 29 figures, 7 tables and 14 references.

Original language: Croatian

Mentor:

Saša Mladenović, Ph.D. *Associate Professor of Faculty of Science, University of Split*

Ocjenjivači:

Saša Mladenović, Ph.D. *Associate Professor of Faculty of Science, University of Split*

Divna Krpan, Ph.D. *Assistant Professor of Faculty of Science, University of Split*

Goran Zaharija, Ph.D. *Assistant Professor of Faculty of Science, University of Split*

Thesis accepted: September 2022

IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam diplomski rad s naslovom Korisnost pretvorbe vremenskih podataka u sliku u svrhu klasifikacije koristeći duboko učenje izradio samostalno pod voditeljstvom izvanrednog profesora dr. sc. Saše Mladenovića. U radu sam primijenio metodologiju znanstvenoistraživačkog rada i koristio literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u diplomskom radu na uobičajen, standardan način citirao sam i povezo s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika

Student

Josip Katić

Sadržaj

UVOD.....	1
1. Skup podataka	2
2. EEG	5
2.1. Moždani valovi.....	5
2.2. Korisnosti EEG-a.....	7
3. Arhitekture neuronskih mreža	9
3.1. Recurrent Neural Network – RNN	9
3.1.1. Long short-term memory – LSTM.....	10
3.1.2. Gated recurrent unit – GRU.....	11
3.2. Convolutional Neural Network – CNN.....	12
3.2.1. Konvolucijski sloj.....	13
3.2.2. Pooling sloj.....	14
3.3. Usporedba – CNN vs RNN	15
4. Algoritmi za transformaciju podataka	17
4.1. GAF – Gramian Angular Field.....	17
4.2. MTF – Markov Transition Field.....	18
4.3. Vizualni prikaz	19
5. Istraživanje.....	20
5.1. Hipoteza.....	20
5.2. Alati	20
5.3. RNN.....	21
5.4. CNN.....	23
5.4.1. GASF.....	25
5.4.2. GADF	28

5.4.3. MTF.....	31
5.5. F mjera.....	34
5.6. Konačna opažanja.....	36
Zaključak	39
Literatura	40
Ilustracije	42
Tablice	44

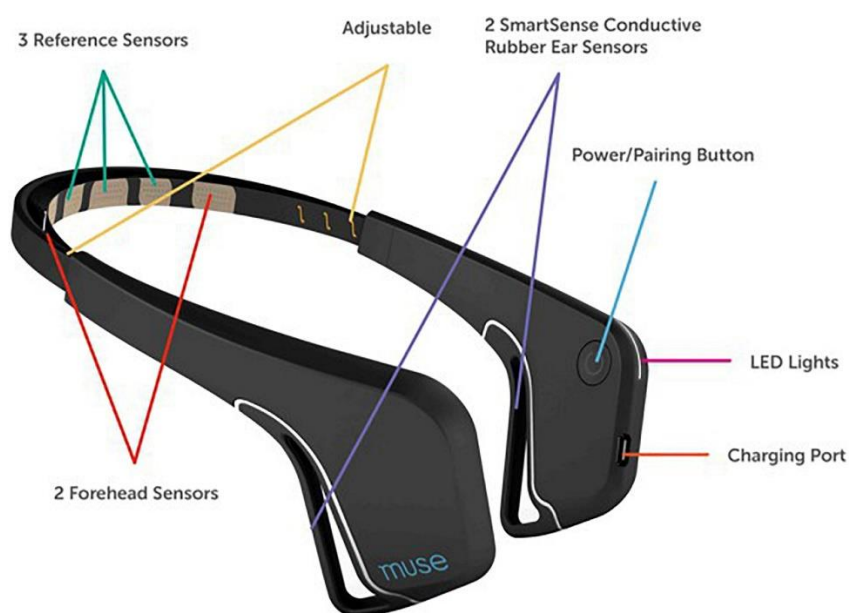
UVOD

Računalstvo je jedna od grana znanosti kod koje se događaju konstantni, znatni i jako brzi napredci koje kao korisnici i inovatori moramo biti spremni iskoristiti da bi mogli unaprijediti kako druga područja tako i samo područje računalstva. Jedna od grana računalstva koja u posljednje vrijeme doživljava jako velike napretke zbog korištenja u raznim industrijama je računalni vid. Trenutno vjerojatno najpopularnija industrija u kojoj se primjenjuje je auto industrija, u svrhu samovozećih automobila koji su u posljednjih nekoliko godina došli do velikih napredaka. Područje računalnog vida je naravno bazirano na grani umjetne inteligencije, točnije rečeno na području dubokog učenja čiji će se koncepti koristiti unutar ovog rada.

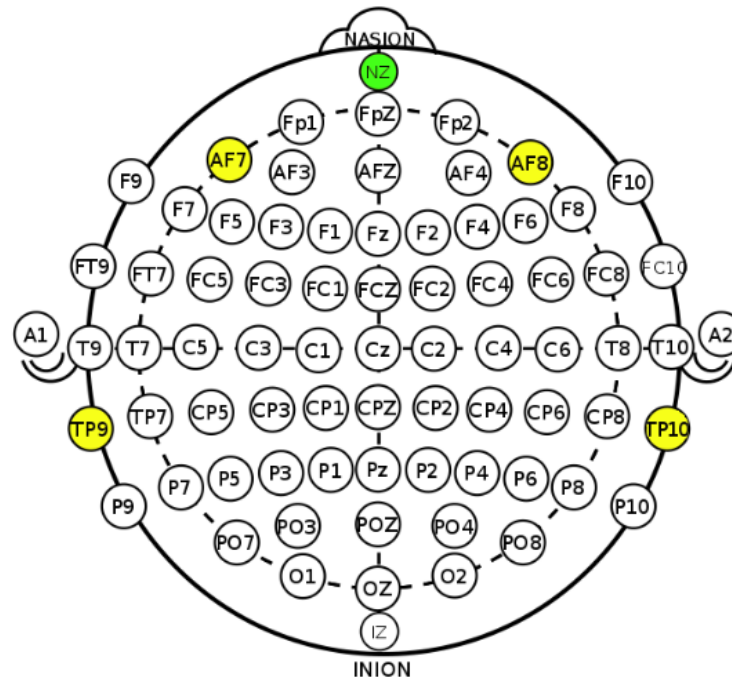
Pitanje na koje tražimo odgovor unutar ovog rada je vezano za neuronske mreže, točnije rečeno za konvolucijske neuronske mreže. Cilj nam je provjeriti njihovu točnost pri klasifikaciji slika koje su dobivene pri transformaciji vremenskih podataka u slike koristeći određene algoritme za pretvorbu podataka. Korišteni algoritmi su GAF, koji je podijeljen na algoritme GASF i GADF, te MTF. Vremenski podatci koji će se koristiti su dobiveni koristeći EEG za mjerenje moždanih valova koji su izabrani zbog svoje nepredvidljivosti i nestabilnosti. Da bi provjerili korisnost pretvorbe takvih podataka u slike uspoređivat ćemo dobivene rezultate iz konvolucijske neuronske mreže sa rezultatima dobivenim koristeći rekurentnu neuronsku mrežu koja je primjerenija za korištenje vremenskih podataka te će se koristiti originalnim skupom EEG podataka.

1. Skup podataka

Skup podataka korišten za istraživanje sadrži EEG podatke moždanih valova preko kojih su se mjerila mentalna stanja (opušteno, neutralno i koncentriranje). Podatci su prikupljeni od četiri osobe (2 muško, 2 žensko) u vremenu od 60 sekundi za svako od tri stanja. Za skupljanje podataka korištena je Muse EEG traka za glavu koja je snimala TP9, AF7, AF8 i TP10 EEG pozicije preko suhih elektroda te koristila jednu od elektroda(NZ) kao referentnu točku. AF7 i AF8 su pozicije na prednjem dijelu lubanje, dok se TP9 i TP10 nalaze na stražnjem dijelu glave. Naravno ovaj broj elektroda bi bio nedovoljan kod pretraga u zdravstvene svrhe gdje se pretežno koristi od 16 do 25 elektroda, a u velikom broju slučajeva u znanstvene svrhe se koristi sve od 100 pa čak do 200 elektroda, i moramo biti svjesni da veći broj elektroda dovodi do boljeg i točnijeg mjerenja.

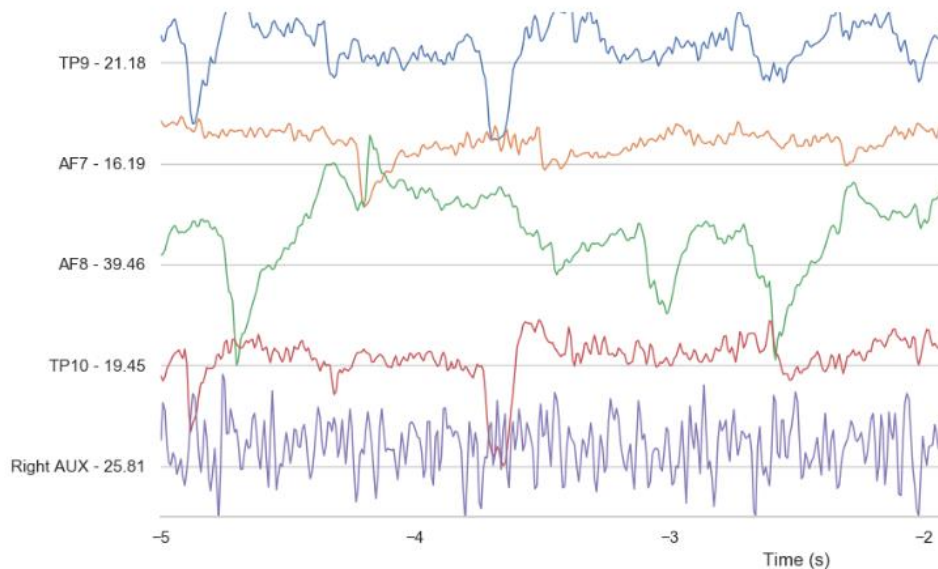


Slika 1 Muse headband



Slika 2 Standardne pozicije elektroda

Na slici osim standardnih pozicija elektroda imamo označene pozicije od 4 elektrode koje se nam dostupne na *Muse headbandu*, označene žutom bojom, te zelenom bojom je označena referentna točka za kalibraciju(NZ).



Slika 3 Primjer EEG signala s Muse senzora

Kod prikupljanja podataka za opušteno stanje subjektu se pustila glazba s niskim tempom i zvukovi dizajnirani u svrhu pomaganja pri meditaciji te su subjekti dobili uputu da opuste

mišiće i naravno odmaraju. Za neutralno stanje uma podatci su se prikupljali na sličan način uz razliku što se subjektima nisu pružili nikakvi stimulansi. Ovo mjerenje se izvršilo prije mjerenja za ostala dva stanja zbog mogućnosti ostajanja trajnijeg efekta od mjerenja opuštenog ili koncentriranog stanja. Za koncentrirano stanje subjekti su trebali pratiti „igru školjki“ (čaše i loptica) u kojoj je loptica sakrivena ispod jedne od tri čaše/školjke kojima su se onda izmjenjivale pozicije. Zadatak subjekta je bio pratiti ispod koje čaše se nalazi loptica.

2. EEG

EEG je tehnika korištena u medicini u svrhu slikovitog prikaza podataka koji su prikupljeni očitavanjem električnih signala s glave koji su proizvedeni od strane mozga. Neke druge tehnike za slikovito prikazivanje podataka su ECG (srce), EMG (mišićne kontrakcije), MEG (mozak), EGG (stomak). EEG je definiran kao električna aktivnost alternirajućeg tipa snimljena s glave nakon što je uhvaćena s metalnim elektrodama i vodljivim medijem. Najveća prednost EEG-a je brzina. Dopušta nam snimanje kompleksnih uzoraka živčanih aktivnosti koje se pojavljuju u djelićima sekunde nakon što se pruži stimulans.

Sustav koji koristimo za snimanje EEG-a se sastoji od

- Elektroda s vodljivim medijem
- Pojačalima s filterima
- A/D konverterima
- Uređajima za snimanje

Elektrode očitavaju signale s površine glave, pojačala podižu signale na razinu na kojoj mogu biti digitalno točni, konverter prevodi analogne signale u digitalne informacije i računalo ili drugi prikladan uređaj se koristi za spremanje i prikazivanje podataka.

2.1. Moždani valovi

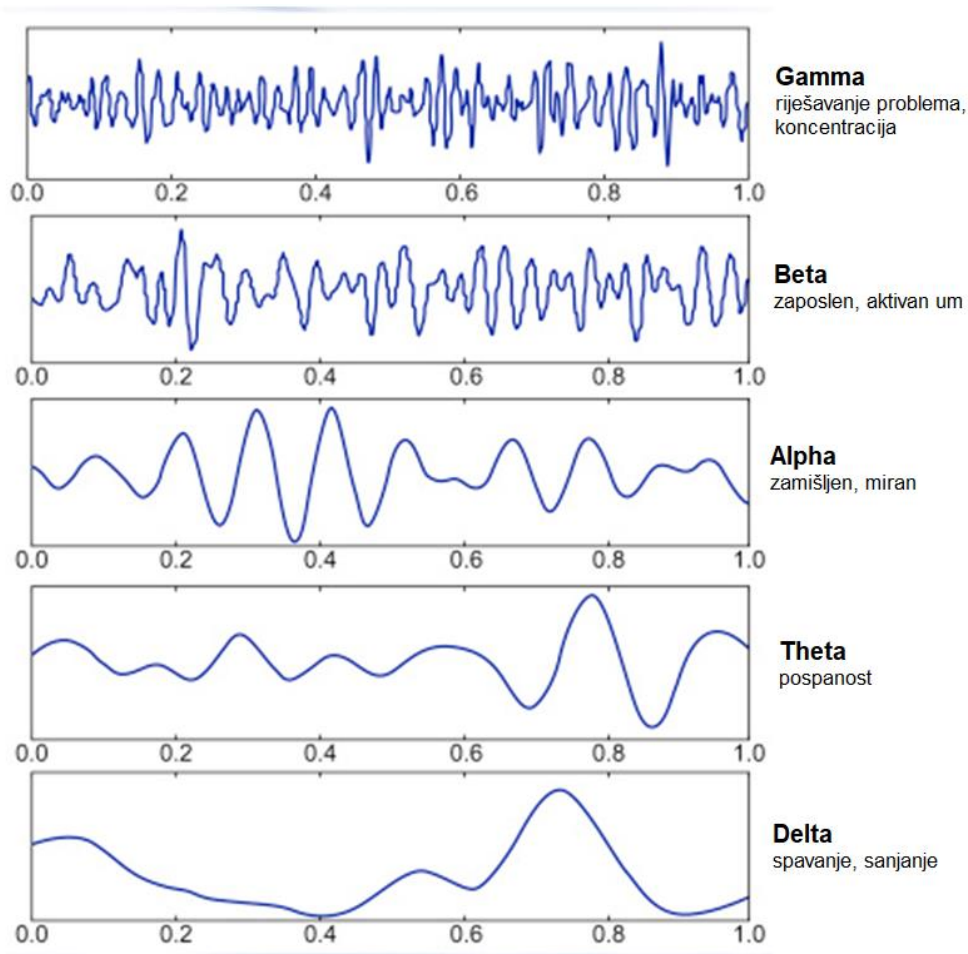
Ljudski mozak sadrži u sebi milione moždanih stanica koje također nazivamo neuronima. Neuroni unutar našeg mozga funkcioniraju tako da međusobno komuniciraju koristeći se električnim impulsima. Naročito zbog komunikacije unutar živčanog sustava neuroni su međusobno povezani na način da formiraju neuronske mreže, a povezani su kroz prolaze koji se nazivaju sinapse koje dopuštaju aktivnosti. Kada se dogodi aktivnost neurona te se pokrene električni impuls od strane jednog neurona prema drugome dolazi do stvaranja valnih uzoraka/valova koje nazivamo moždanim valovima.

Također postoji razlika među moždanim valovima prema aktivnosti koja ih pokreće, tj. prema stanju mozga za vrijeme te aktivnosti. Primjer toga je što možemo očekivati različito ponašanje mozga i vrstu moždanog vala ako osoba se primjerice odmara ili spava u kontrastu sa stanjem kada je osoba anksiozna ili u dubokoj koncentraciji. Te različite valove mjerimo prema različitim frekvencijama koje posjeduju, a postoje 5 tipova valova.

Oznaka/naziv	Frekvencija	Stanje mozga
Gamma	>35 Hz	Koncentracija
Beta	12–35 Hz	Anksioznost dominantna, aktivno, eksterna pažnja, opušteno
Alpha	8–12 Hz	Jako opušteno, pasivna pažnja
Theta	4–8 Hz	Duboka opuštenost, unutrašnji fokus
Delta	0.5–4 Hz	Spavanje

Tablica 1 Podatci o moždanim valovima

Za prikupljanje osnovnih uzoraka moždanih valova od subjekata se traži da zatvore oči te se opuste. Dobiveni uzorci kod mjerenja su pretežno sinusnog oblika te se obično mjere od vrha do vrha koji su najčešće u rasponu od 0.5 do 100 μ V u amplitudi.



Slika 4 Primjer moždanih valova

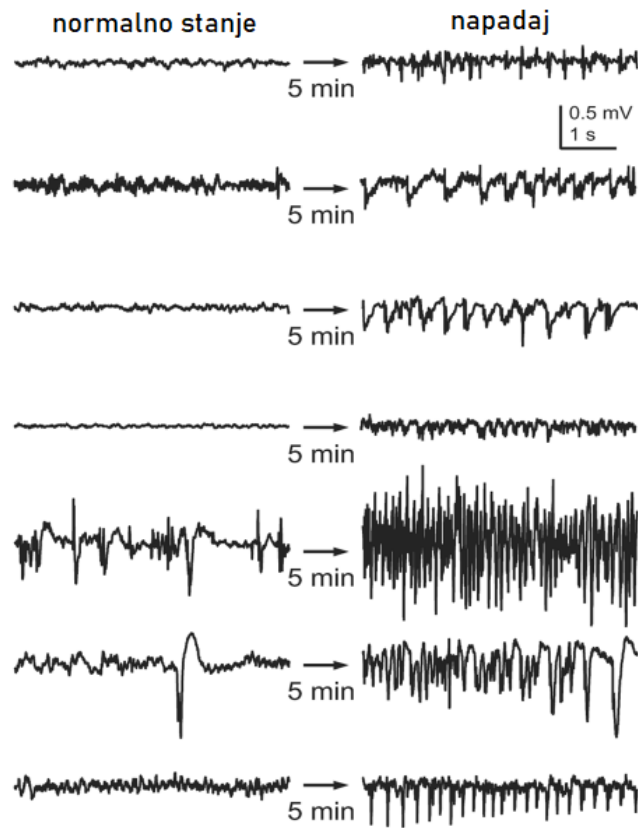
2.2. Korisnosti EEG-a

EEG možemo koristiti da bi pronašli promjene u moždanoj aktivnosti koje bi mogle biti od pomoći pri dijagnozi poremećaja mozga, a najbolji primjer takvog poremećaja je epilepsija ili neki slični poremećaji koji uzrokuju napadaje.

Korištenje EEG-a u medicinske svrhe kod ljudi i životinja

- Nadgledanje budnosti, koma i smrt mozga
- Lociranje područja ozljede zbog ozljede glave, moždanog udara, tumora i slično
- Istraživanje epilepsije i lociranje podrijetla napadaja
- Testiranje utjecaja lijekova
- Nadgledanje razvoja mozga

- Istraživanje poremećaja sna



Slika 5 Očitavanje napadaja preko EEG-a

Pošto je EEG procedura ne invazivna i bezbolna, široko se koristi za istraživanje kognitivnih procesa kao što su percepcija, memorija, pozornost, jezik i emocije kod odraslih i djece.

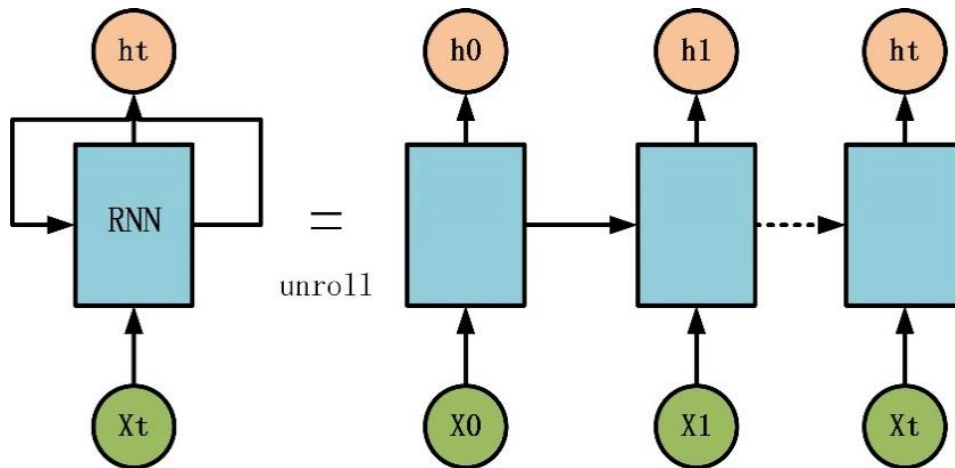
3. Arhitekture neuronskih mreža

Iako postoji veliki broj arhitektura neuronskih mreža unutar ovog rada su se uspoređivale samo dvije. Prva arhitektura je rekurentna neuronska mreža (RNN) s kojom smo promatrali i radili nad izvornim vremenskim podacima dobivenim preko EEG-a, a druga arhitektura je konvolucijska neuronska mreža (CNN) koja je dobila transformirane EEG podatke iz vremenskog oblika u oblik slika.

3.1. Recurrent Neural Network – RNN

RNN je bilo koja neuronska mreža čiji neuroni međusobno šalju povratne signale, na taj način izlaz nekog čvora naknadno može utjecati na ulaz tog istog čvora. Ovo omogućava RNN-u vremenski dinamično ponašanje, tj. stvorili smo neuronsku mrežu koja šalje ulazne podatke za buduće korištenje. RNN-ovi mogu koristiti svoje unutarnje stanje, memoriju, za obradu nizova ulaza promjenjive duljine. Zbog ovog svojstva su primjenjive za korištenje nad zadacima kao što su prepoznavanje govora ili prepoznavanje povezanog ručno pisanog teksta.

Pojam *recurent neural network* se koristi za vrstu neuronskih mreža koje imaju beskonačni odgovor impulsa (*infinite impulse response* - IIR), a to je svojstvo koje govori da nakon neke vremenske točke vrijednost impulsa ne dostiže nulu nego nastavlja na ne određeno. Kod CNN-ova za razliku od RNN-ova imamo situaciju ograničenog impulsnog odgovora.



Slika 6 Slikoviti prikaz RNN arhitekture

Postoje dva popularna i efikasna RNN modela koji funkcioniraju jako dobro, a to su *long short-term memory* i *gated recurrent unit*.

3.1.1. Long short-term memory – LSTM

LSTM je memorijska jedinica kod neuronskih mreža koja koristi 3 vrata za upravljanje sadržajem memorije. Ova vrata su jednostavne logičke funkcije težinskih suma preko kojih neuronska mreža uči, pamti i prisjeća se onoga što joj je potrebno.

Ulazna vrata (1) i vrata za zaboravljanje (2) upravljaju stanjem ćelije (4) koje predstavlja dugoročno pamćenje. Izlazna vrata (3) stvara izlazni vektor ili skriveno stanje (5) koje predstavlja memoriju koja se koristi. Ovaj memorijski sustav omogućava mreži pamćenje na duže vrijeme što je bio nedostatak kod prethodnih rekurentnih neuronskih mreža.

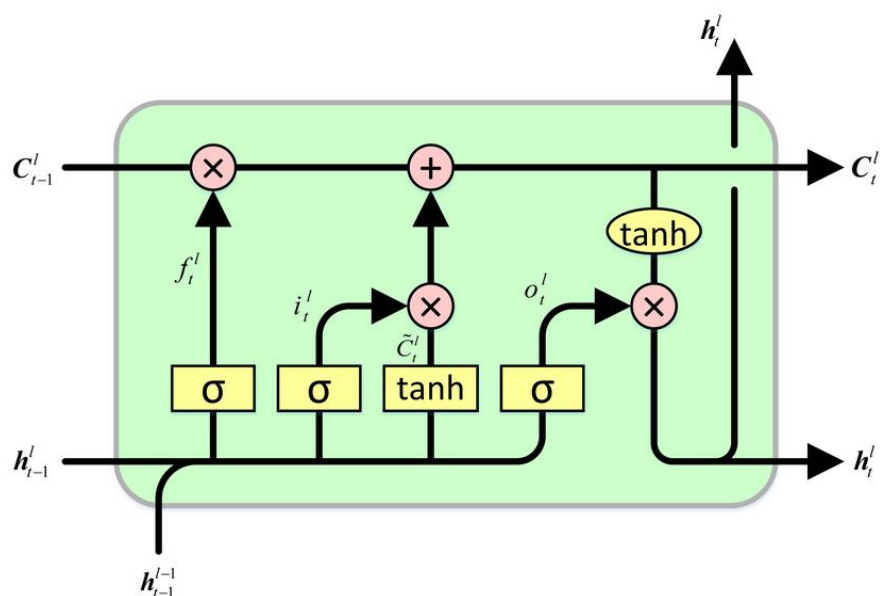
$$i_t = \text{sigmoid}(W_i x_t + U_i h_{t-1} + b_i) \quad (1)$$

$$f_t = \text{sigmoid}(W_f x_t + U_f h_{t-1} + b_f) \quad (2)$$

$$o_t = \text{sigmoid}(W_o x_t + U_o h_{t-1} + b_o) \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \odot \tanh(c_t) \quad (5)$$



Slika 7 LSTM osnovna arhitektura

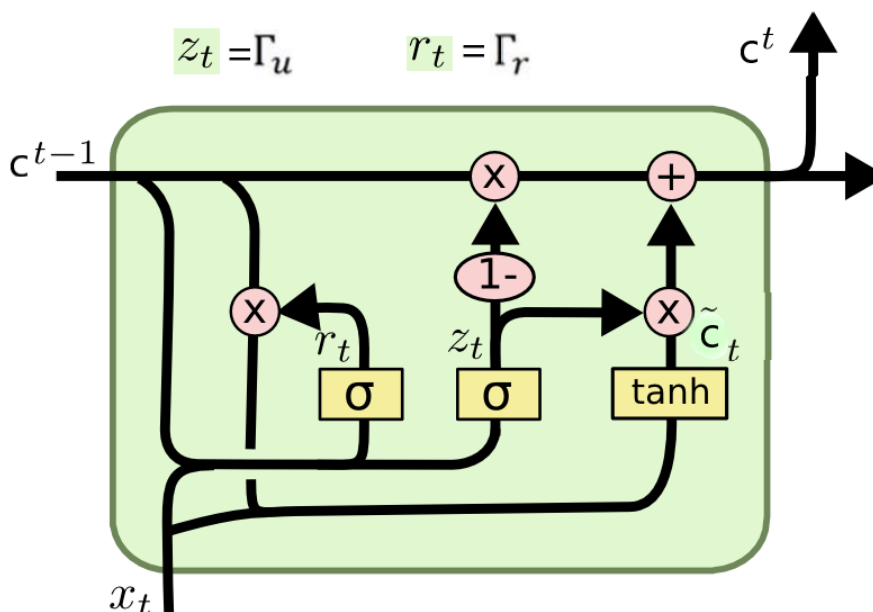
3.1.2. Gated recurrent unit – GRU

GRU je donekle pojednostavljena verzija LSTM-a koja ima istu zadaću unutar neuronske mreže. Glavna razlika GRU-a naprema LSTM-u je broj vrata i težina, gdje kao što je već rečeno GRU je donekle jednostavniji. Sadrži samo 2 vrata za razliku od LSTM-a koji sadrži 3. Ne sadrži izlazna vrata što znači da kontrola nad memorijskim sadržajem je ne postojeća. Prva vrata su ažurirajuća vrata (6) koja kontroliraju tok informacija od prethodnih aktivacija te također i dodavanje novih informacija (8). Ona odlučuju jeli stanje se treba ažurirati prema trenutnoj aktivnoj vrijednosti ili ne. Druga vrata su resetirajuća (7) koja odlučuju jeli prethodno stanje ćelije bitno ili ne.

$$z_t = \text{sigmoid}(W_z x_t + U_z h_{t-1} + b_z) \quad (6)$$

$$r_t = \text{sigmoid}(W_r x_t + U_r h_{t-1} + b_r) \quad (7)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (8)$$

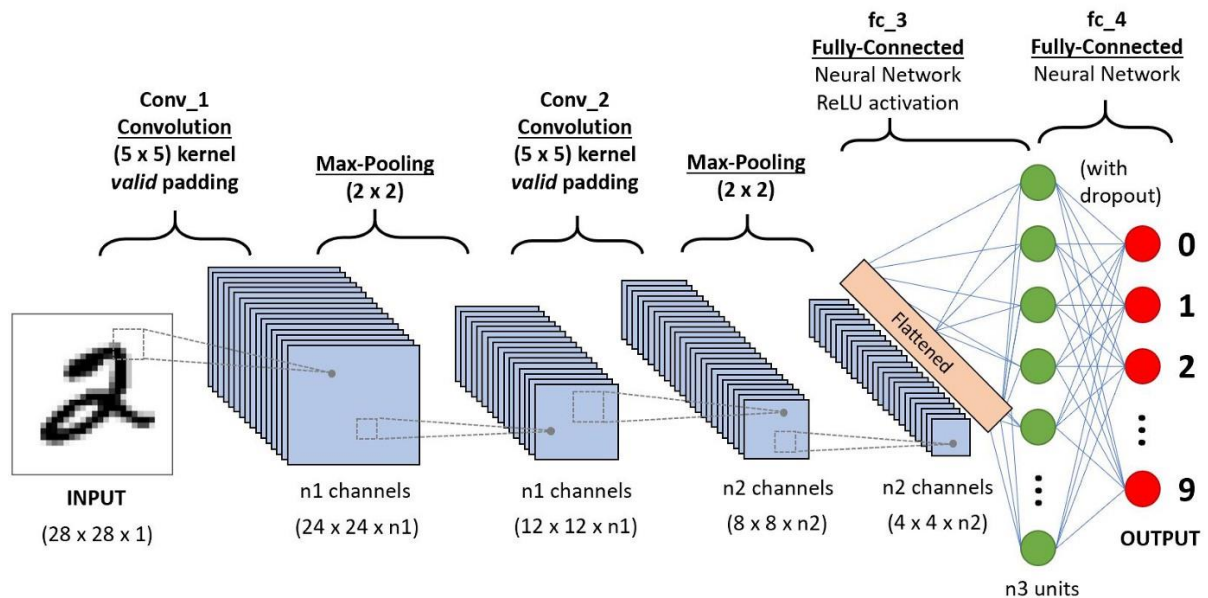


Slika 8 GRU osnovna arhitektura

3.2. Convolutional Neural Network – CNN

Konvolucijske neuronske mreže su klasa neuronskih mreža koja se najčešće koristi za prepoznavanje uzoraka kod slika. Ovo nam dopušta da kodiramo značajke koje su specifične za slike unutar arhitekture te time stvaramo neuronsku mrežu koja je prikladnija za zadatke fokusirane na slike uz daljnje smanjenje broja parametara potrebnih za postavljanje modela.

Jedna od ključnih razlika kod CNN arhitekture je što su neuroni od kojih su slojevi mreže građeni organizirani u tri dimenzije. Prve dve dimenzije su dimenzije slika, tj. visina i šira slike, a treća dimenzija je dubina. U ovom kontekstu dubina nam ne predstavlja broj slojeva unutar neuronske mreže nego predstavlja treću dimenziju aktivacijskog volumena. U praksi ovo bi izgledalo na način da su dimenzije ulaznog volumena $64 * 64 * 3$ (visina, širina i dubina) vodeći do konačnog izlaznog sloja koji ima dimenzije $1 * 1 * n$ (n predstavlja mogući broj klasa) kako bi stisnuli dimenzije početnog ulaza u manji volumen klasnih rezultata kroz dimenziju dubine.



Slika 9 Slikoviti prikaz CNN arhitekture

Osnovne funkcionalnosti CNN modela možemo razdvojiti u četiri ključna područja

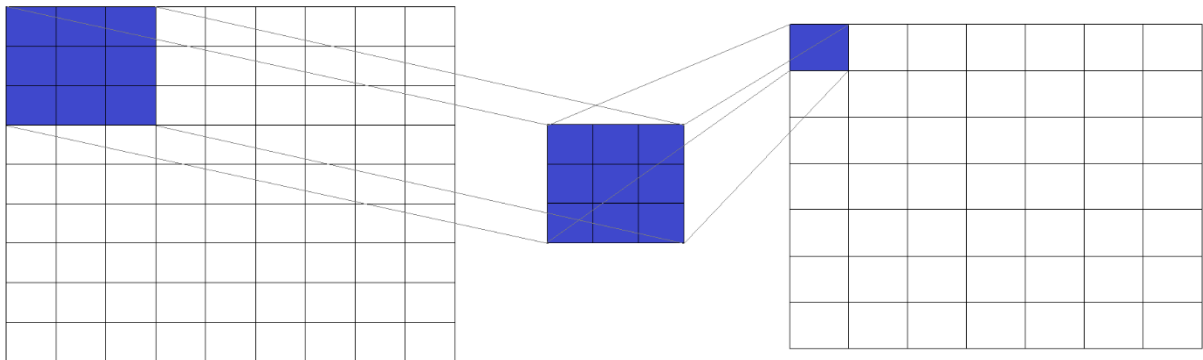
1. Ulazni sloj sadrži piksel vrijednosti korištene slike
2. Konvolucijski sloj će odrediti izlaz neurona, koji su povezani na regije na ulazu, kroz izračune skalarnog produkta između njihovih težina i veličina povezanih regija
3. *Pooling* sloj odrađuje smanjivanje uzorka kroz prostornu dimenziju danog ulaza i time dalje smanjiva broj parametara
4. Potpuno povezani sloj na kraju odrađuje akcije za dobivanje rezultata klasifikacija kao i kod drugih ANN arhitektura

3.2.1. Konvolucisjki sloj

Kao što se može pretpostaviti prema imenu konvolucijki sloj ima važnu ulogu u CNN arhitekturi. Sloj se bazira na korištenju kernela ili filtera koji mogu učiti. Ovi filteri su uobičajeno malih dimenzija, ali pokrivaju cijelu dubinu ulaza. Konvolucijski sloj nam je jedan od načina za rješavanje problema veličine podataka. Treniranje klasičnog ANN modela na ulazu kao što je slika rezultira s modelom koji su previše kompleksni i veliki za efikasno treniranje.

Filter unutar konvolucijskog sloja funkcionira na način da prolazi kroz dvodimenzionalnu matricu podataka, koja predstavlja korištene slike, u njoj cijelosti te nad podacima koji

se nalaze u filteru izvršava množenje tih elemenata. Kao rezultat množenja dobivamo njihovu vrijednost u veličini jednog izlaznog piksela. Filter takvu operaciju izvršava nad svakom pozicijom unutar matrice preko koje prođe te ćemo kao izlaz dobiti potpuno novu matricu manjih dimenzija.



Slika 10 Vizualizacija kovolucijskog sloja

Konvolucijski sloj također može znatno smanjiti kompleksnost modela kroz optimizaciju izlaza. Ovo je prilagođeno koristeći tri parametra: dubina, iskorak i postavljanjem nulte postave (eng. *zero-padding*). Kroz korištenje ove tehnike utječemo na prostorne dimenzije izlaza kovolucijskog sloja, a da bi ih izračunali koristimo formulu

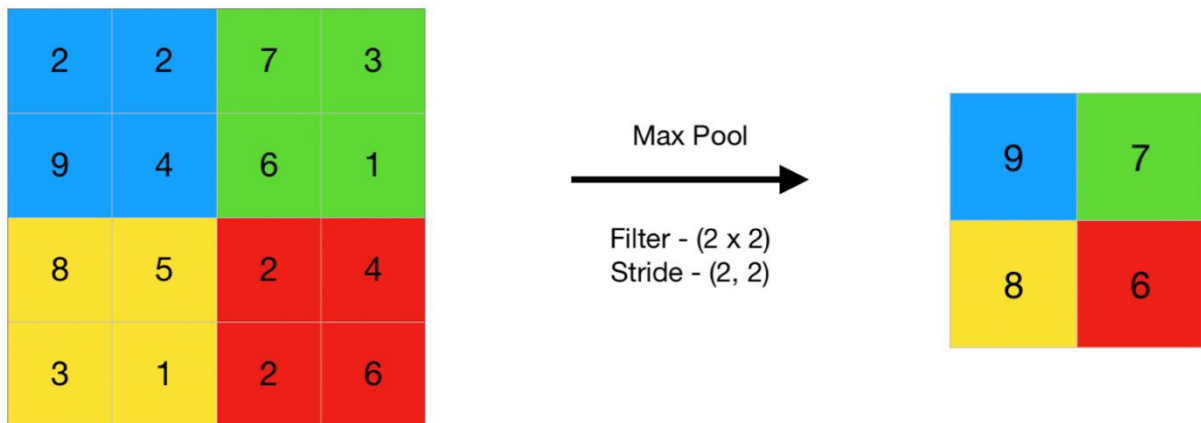
$$\frac{(V - R) + 2Z}{S + 1} \quad (9)$$

V predstavlja volumen ulaza (visina * širina * dubina), R veličinu receptivnog polja, Z je postavljena količina *zero-paddinga* i S se odnosi na iskorak. U slučaju da rezultat ove jednadžbe nije jednak cijelom broju onda imamo krivo postavljen iskorak. Tada neuroni neće biti u mogućnosti primjereno se postaviti unutar danog ulaza.

3.2.2. Pooling sloj

Cilj *pooling* sloja je postepeno smanjiti dimenzije reprezentacije te s time smanjiti broj potrebnih parametara i kompleksnosti modela. *Pooling* sloj djeluje nad svakom aktivacijskom mapom s ulaza, te skalira dimenzije koristeći „MAX“ funkciju. Kod većine CNN modela ovo imamo u obliku *max-pooling* sloja s filterom dimenzija 2*2 primijenjenim s iskorakom 2 preko prostorne dimenzije ulaza. S ovim smanjujemo aktivacijsku mapu na 25% od originalne veličine dok zadržavamo dubinu.

Zbog destruktivne prirode *pooling* sloja uobičajeno se koriste samo dvije metode *max-pooling*. Uobičajeno su iskorak i filter postavljeni na 2×2 što omogućava sloju da se raširi preko cijele prostorne dimenzije ulaza. Također moguće je koristiti *pooling* s preklapanjem gdje je iskorak postavljen na 2 dok je filter postavljen na 3. Zbog destruktivnosti *pooling* uobičajeno se ne koristi filter veći od 3 zbog velike mogućnosti smanjivanja kvalitete modela.



Slika 11 Vizualizacija Max Pool (2×2)

Na slici poviše vidimo primjer skaliranja matrice koristeći *max-pool* koji koristi dimenzije iskoraka i filtera 2×2 . Iz početne matrice koja je podijeljena na četiri dijela s četiri filtera za dobivanje nove skaliranje matrice iz svakog od područja uzimamo samo onu najdominantniju, tj. najveću vrijednost. Rezultat toga je što dobivamo manju matricu ali onu koja sadrži samo vrijednosti koje su se istakle unutar prethodne matrice.

3.3. Usporedba – CNN vs RNN

CNN su preferirani izbor kod interpretacije vizualnih podataka, oskudnih podataka ili podataka koji ne dolaze u slijedu.

Za razliku od CNN, RNN su dizajnirane za prepoznavanje slijednih i vremenskih podataka. Odrađuju bolja predviđanja s obzirom na redoslijed podataka jer su povezani s prethodnim ili sljedećim čvorom.

	CNN	RNN
Arhitektura	<i>Feed-forward</i> neuronska mreža koja koristi filtere i <i>pooling</i>	Rekurentna mreža koja pohranjuje rezultate natrag u mrežu
Ulaz/Izlaz	Veličina ulaza i izlaznog rezultata su fiksirani (npr. mreža dobije sliku fiksirane veličine i na izlazu ih kategorizira i daje količinu pouzdanosti u tu kategorizaciju)	Veličina ulaza i rezultat izlaza može varirati (npr. na ulaz dobije različite tekstove te za izlaz ispiše prevode – izlazna rečenica može imati više ili manje riječi)
Idealni scenarij korištenja	Prostorni podatci (slika)	Vremenski/sekvencijalni podatci (npr. tekst ili video)
Korištenje	Identifikacija i klasifikacija slika, detekcija lica, analiza podataka u medicini, otkrivanje lijekova, analiza slika	Prevođenje teksta, procesiranje prirodnih jezika, ekstrakcija entiteta, razgovorna inteligencija, analiza sentimenta, analiza govora

Tablica 2 CNN i RNN razlike

4. Algoritmi za transformaciju podataka

Unutar ovog istraživanja za pretvorbu vremenskih podataka dobivenih putem EEG-a u oblik slika, oblik prikladan za rad s konvolucijskim neuronskim mrežama, korištena su dva algoritma. Prvi algoritam je *Gramian Angular Field* (GAF) koji sadrži dvije varijacije, *Gramian Angular Summation Field* (GASF) i *Gramian Angular Difference Field* (GADF), a drugi korišteni algoritam je *Markov Transition Field* (MTF).

4.1. GAF – Gramian Angular Field

GAF kodira vremenske podatke u sliku preko matrice temeljene na polarnim koordinatama te može sačuvati apsolutnu temporalnu korelaciju. Početni x iz vremenskih podataka je najprije normaliziran da bude između 0 i 1, definirano jednažbom

$$\tilde{x}_0^t = \frac{x(t) - \min(x)}{\max(x) - \min(x)} \quad (10)$$

Nakon prilagodbe podataka koriste se kut kosinusa i vremenski pečat da bi se kodirali podatci u polarne koordinate. Od gornjeg lijevog kuta do desnog donjeg kuta pozicije slike odgovaraju izvornim vremenskim podacima te su simetrični po dijagonali. Zbog ove činjenice podatci iz slike se mogu vratiti u oblik originalnog vremenskog signala.

GAF iz vremenskih podataka može generirati dvije slike koristeći dvije različite jednažbe.

Prva varijacija je GASF – *Gramian Angular Summation Field*

$$GASF = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix} \quad (11)$$

$$GASF = \tilde{x} * \tilde{x} - \sqrt{I - \tilde{x}^2} * \sqrt{I - \tilde{x}^2} \quad (12)$$

a druga je GADF – *Gramian Angular Difference Field*

$$GADF = \begin{bmatrix} \sin(\varnothing_1 + \varnothing_1) & \dots & \sin(\varnothing_1 + \varnothing_n) \\ \sin(\varnothing_2 + \varnothing_1) & \dots & \sin(\varnothing_2 + \varnothing_n) \\ \vdots & \ddots & \vdots \\ \sin(\varnothing_n + \varnothing_1) & \dots & \sin(\varnothing_n + \varnothing_n) \end{bmatrix} \quad (13)$$

$$GADF = \sqrt{I - \widetilde{x}^2} * \widetilde{x} - \widetilde{x} * \sqrt{I - \widetilde{x}^2} \quad (14)$$

Razlika između GASF i GADF je u trigonometrijskoj funkciji koju koriste, GASF je baziran na kosinu funkciji dok je GADF baziran na sinus funkciji.

4.2. MTF – Markov Transition Field

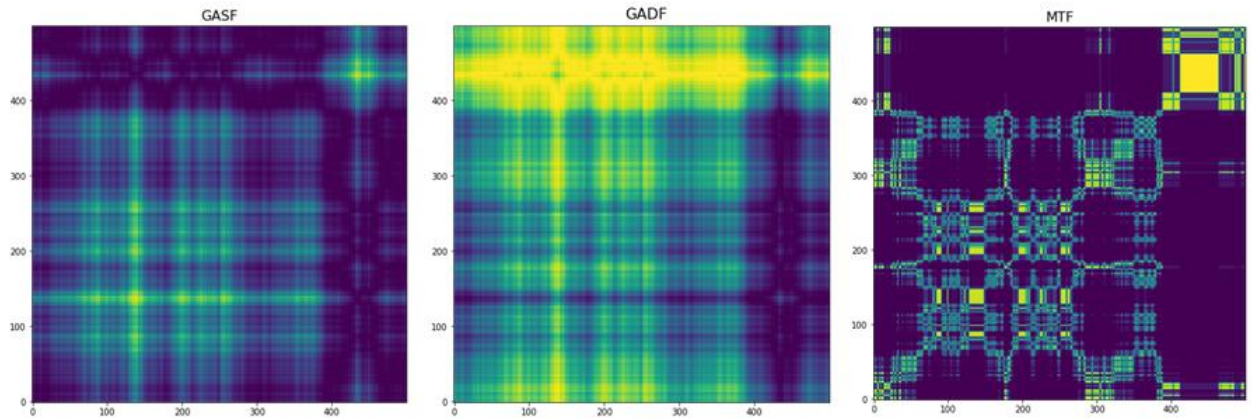
MTF koristi Markovljeve prijelazne vjerojatnosti da bi zadržali detalje u vremenskoj domeni. MTF je sastavljen od Markovljevih prijelaznih vjerojatnosti M_{ij} od kvantilnog biniranja q_i se premješta na q_j , na vremenskoj oznaci i i j . Ako uzmemo vremenske podatke $x = x(1), x(2), \dots, x(T)$ i kvantile $Q = q_1, q_2, \dots, q_j$. Veličina Q utječe na veličinu Markovljeve prijelazne matrice

$$M_{ij} = \begin{bmatrix} w_{ij} | x(1) \in q_i, x(1) \in q_j & \dots & w_{ij} | x(1) \in q_i, x(n) \in q_j \\ w_{ij} | x(2) \in q_i, x(1) \in q_j & \dots & w_{ij} | x(2) \in q_i, x(n) \in q_j \\ \vdots & \ddots & \vdots \\ w_{ij} | x(n) \in q_i, x(1) \in q_j & \dots & w_{ij} | x(n) \in q_i, x(n) \in q_j \end{bmatrix} \quad (15)$$

MTF može sačuvati detalje u vremenskom rasponu, ali kako je transformirana matrica stvorena preko vjerojatnosti kretanja elemenata ne možemo se vratiti na izvorne vremenske podatke kao i kod GAF metode. Za obje metode, GAF i MTF, pretvorene metode se mogu pokazati preko mape boja gdje crvene boje predstavljaju veću vrijednost a plave boje manju vrijednost.

4.3. Vizualni prikaz

Rezultati pretvorbe nasumično generiranog vremenskog niza koristeći prethodno navedene algoritme



Slika 12 GASF, GADF, MTF rezultat pretvorbe

Za demonstraciju je korišten vremenski niz podataka od 1000 nasumično generiranih brojeva nad kojima su korišteni prethodno navedeni algoritmi da bi ih pretvorili u slike dimenzija 500*500.

5. Istraživanje

Cilj provedenog istraživanja je bila provjera sposobnosti do kovolucijskih neuronskih mreža za rad nad podacima koji nisu za njih primjereni. Iako su se konvolucijske neuronske mreže već prije koristile u medicinske svrhe, situacije u kojima su se koristile su bile primjerenije. Veliki broj problema na koji se može naići u medicini se može svesti na podatke u obliku slika, ali što ako trebamo obraditi vremenske podatke i u ovom slučaju EEG podatke koji su jako nepredvidljivi. Jesu li konvolucijske neuronske mreže došle do stadija da ih možemo koristiti umjesto rekurentnih neuronskih mreža koje su za rad nad vremenskim podacima znatno primjerenije. Jedan od razloga za ovo razmišljanje je što kroz proteklih nekoliko godina velika količina resursa i znanja unutar računalne industrije se posvetio na područje računalnog vida. Naravno veliki razlog tome su samo vozeći automobili koji su kao i kod specijaliziranih područja također i u svakodnevnom životu postigli iznimne razine popularnosti. Iako su nam samo vozeći osobni automobili i dalje donekle van dometa, područje računalnog vida je doživjelo napretke.

5.1. Hipoteza

Konvolucijske neuronske mreže su u protekli niz godina doživjele znatne napretke, te mogu postići približno jednake rezultate pri klasifikaciji vremenskih podataka kao i rekurentne neuronske mreže.

5.2. Alati

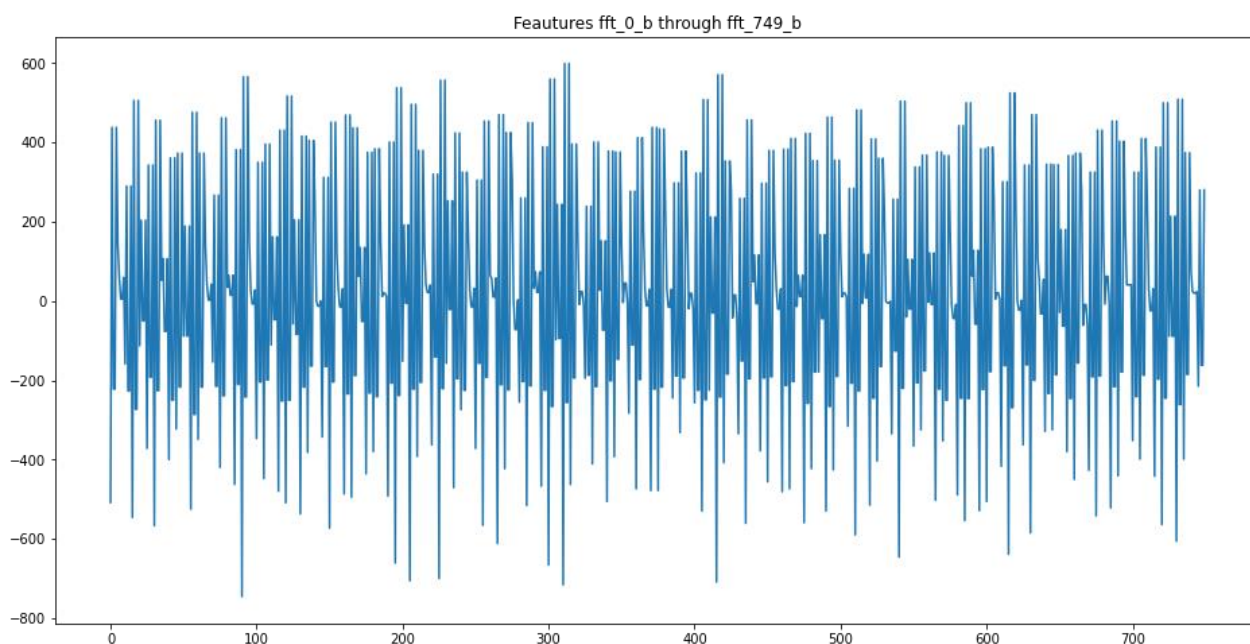
Za čitavi dio izrade projekta koji je bio vezan za kodiranje korišten je programski jezik Python, te njegove biblioteke. U svrhu izrade rekurentne i konvolucijske neuronske mreže korištena je biblioteka *tensorflow* te preko *pyts.image* su uvedeni algoritmi *GramianAngularField* i *MarkovTransitionField*. Također okruženje unutar kojeg se kod pisao je Google Colab.

Početni podatci koji su korišteni za treniranje i testiranje modela su preuzeti iz drugog istraživačkog rada. Preuzeti podatci sadrže EEG mjerenja moždanih valova koja su

prikupljena koristeći Muse traku za glavu, te su procesirana od strane originalnih istraživača na način da su izvukli sve potrebne i korisne informacije za istraživanje.

5.3. RNN

Nakon pronalaska potrebnog skupa podataka sljedeći korak je naravno rad nad njima, a u ovom slučaju je to korištenje podataka za treniranje i testiranje RNN modela. Naravno prije početka korištenja podataka na RNN modelu te podatke je trebalo dodatno pripremiti i proučiti u kakvom obliku su nam dostupni. Skup podataka koji koristimo sadrži 2132 retka i 2549 stupaca od kojih jedan sadržava vrijednosti oznaka (labela) da možemo znati koji su koji podatci. Iako smo već prije definirali kod skupa podataka da se pri mjerenju promatrala 3 stanja opušteno, neutralno i koncentracija, ovdje su spremljeni pod malo drukčijim nazivima. Umjesto opušteno, neutralno i koncentracija imamo oznake negativno, neutralno i pozitivno. Za svaku od tih vrijednosti imamo pretežno jednaku količinu podataka. Za negativnu i pozitivnu oznaku obje imaju 708 redova dok kod neutralne imamo 716 redova.



Slika 13 Odsječak vremenskih podataka

Prije kretanja na izradu i rad s modelom RNN arhitekture potrebno je bilo odvojiti stupac s oznakama, svakoj oznaci zbog jednostavnosti pridodati brojčanu vrijednost (negativno: 0,

neutralno: 1, pozitivno: 2), te konačno podijeliti izmiješani skup podataka na skup za treniranje (pridodano 70% podataka iz cijelog skupa) i skup za testiranje.

Konačno možemo graditi i model naše rekurentne neuronske mreže.

```
inputs = tf.keras.Input(shape=(RNN_x_train.shape[1],))

expand_dims = tf.expand_dims(inputs, axis=2)
gru = tf.keras.layers.GRU(256, return_sequences=True)(expand_dims)
flatten = tf.keras.layers.Flatten()(gru)
outputs = tf.keras.layers.Dense(3, activation='softmax')(flatten)

RNN_model = tf.keras.Model(inputs=inputs, outputs=outputs)

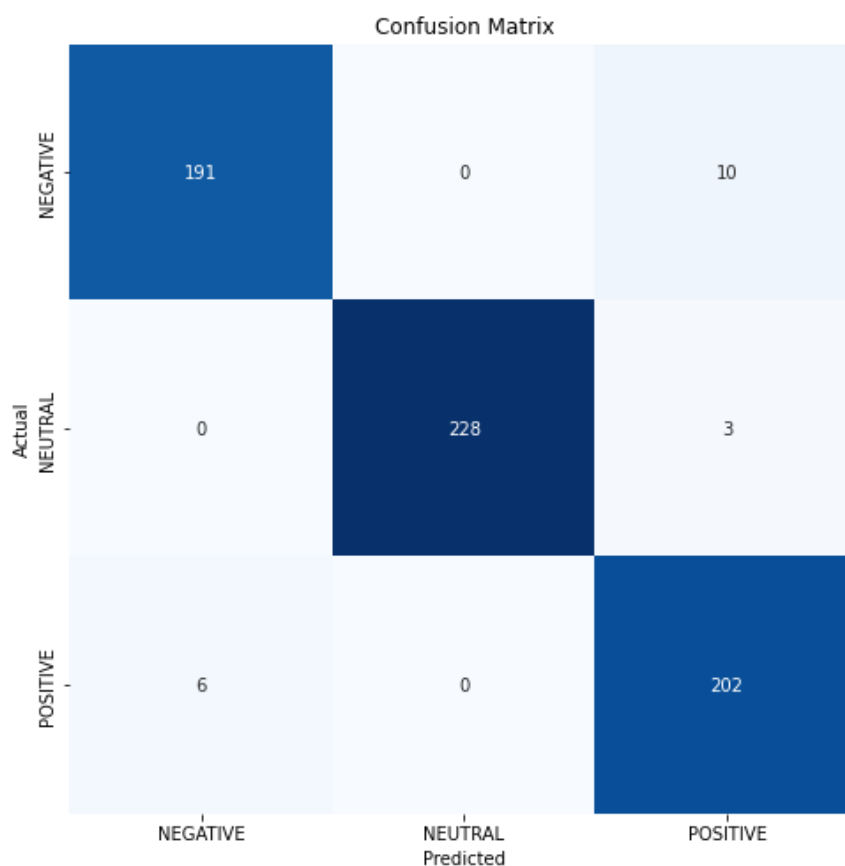
RNN_model.compile(
    optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy']
)
```

Slika 14 RNN model

Unutar modela koristimo GRU sloj za koji je određeno 256 jedinica i koristimo *return_sequences* što znači da ćemo dobiti veći broj podataka u obliku dvodimenzionalnog niza. Zbog toga što nam GRU sloj vraća dvodimenzionalni niz u sljedećem koraku moramo prilagoditi te podatke da budu u obliku jednodimenzionalnog vektora koristeći *Flatten* te konačno za kraj nakon *Flatten* sloja dolazimo do izlaznog sloja.

Nakon izrade modela pozivamo *compile* s *adam* optimizatorom i *sparse_categorical_crossentropy* koji je najbolja opcija kada radimo s problemima višestruke klasifikacije te prelazimo na treniranje modela.

Za treniranje modela korišteno je samo 5 epoha, a konačnom evaluacijom dobili smo da je rezultat točnosti jednak 97,031%



Slika 15 Matrica konfuzije – RNN

Preko matrice konfuzije možemo vidjeti kod kojih predviđanja je naš RNN model dolazio do zabune, tj. pogreške pri klasifikaciji kojem stanju zapravo promatrani podatak pripada. Možemo vidjeti da se najveća pogreška pojavljivala u situaciji kada su podatci predstavljali negativno stanje, ali model ih je klasificirao kao da se radi o pozitivnom stanju. Također možemo vidjeti da se najmanja sveukupna pogreška pojavila kod klasifikacije neutralnih podataka gdje imamo samo tri greške gdje je model umjesto klasificiranja podataka kao neutralne klasificirao kao pozitivne.

5.4. CNN

Za razliku od primjene RNN modela za CNN model smo prvo trebali promijeniti oblik/tip podataka. RNN je tip arhitekture koji je pogodan za rad s vremenskim podacima kao što su korištena mjerenja EEG-a, ali CNN arhitektura je prilagođenija u smjeru računalnog vid, tj. rad sa slikama. Za pretvorbu dostupnih vremenskih podataka u prikladan oblik slika korišteni su već prethodno navedeni algoritmi GAF (GASF i GADF) i MTF. Iako su

korišteni različiti algoritmi za pretvorbu podataka jedan te isti CNN model je korišten za sva tri skupa podataka jer se ipak radi o istom tipu podataka, tj. slikama.

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(2, 2), input_shape=(219, 217, 4),
                activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Conv2D(64, kernel_size=(2, 2), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(3, activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam', metrics=['accuracy'])
```

Slika 16 CNN model

Prvotno što radimo je pozivamo funkciju *Sequential* jer radimo linearni stog slojeva u modelu. Bitna dva sloja koja koristimo u modelu su Conv2D i MaxPool2D. Konvolucijski 2d sloj ili kraće napisano Conv2D je najčešći oblik konvolucije koji se koristi unutar 2d konvolucijskih slojeva.

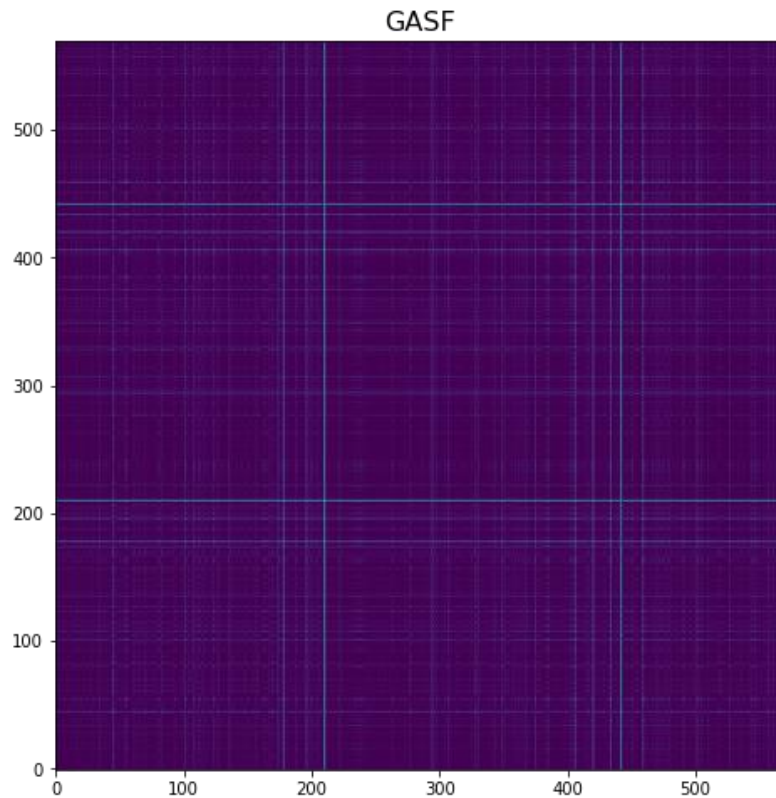
Prije dovođenja podataka do izlaznog sloja kao i kod RNN modela pozivamo *Flatten* da bi prilagodili oblik iz dvodimenzionalnog niza u jednodimenzionalni vektor, te također prije samog izlaznog sloja prolazimo kroz 2 skrivena sloja.

Konačno nakon izrade modela kao i kod RNN modela pozivamo *compile* koji je izgrađen kao i u prethodnom slučaju s *adam* optimizatorom i *sparse_categorical_crossentropy*.

Prije prelaska na treniranje i testiranje CNN modela koristeći slike dobivene s prethodno navedenim algoritmima za pretvorbu bitno je napomenuti da zbog ograničenja nametnutih od snage računala nisu iskorišteni svi postojeći podatci, odnosno slike. Kroz pretvorbu podataka iz vremenskog oblika u slike preko svakog algoritma se generirale 7644 slike (sva od tri postojeća stanja su zasebno imali po 2548 slika), ali od tog kompletnog broja za treniranje i testiranje modela je iskorišteno ukupnih 3000 slika (1000 slika za svako od tri postojeća stanja).

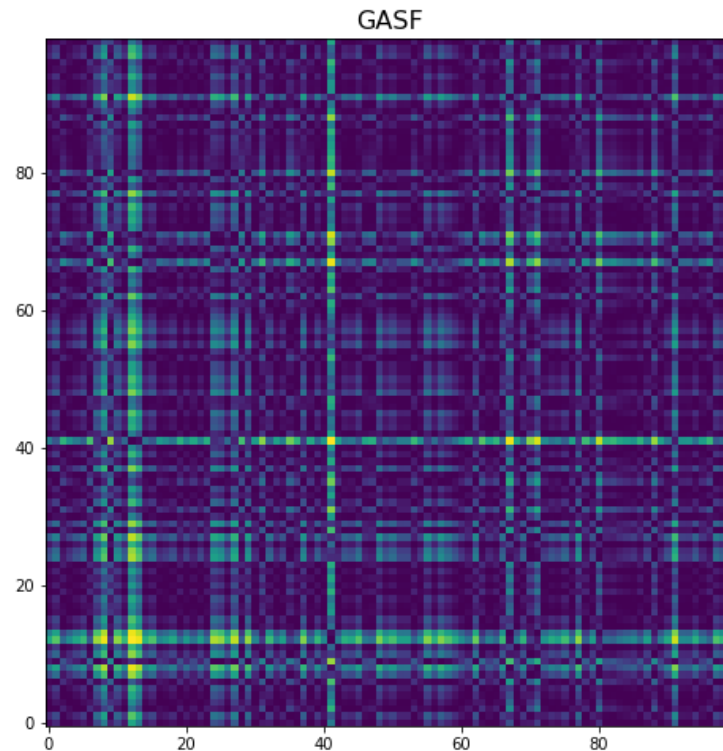
5.4.1. GASF

Prvi algoritam korišten za pretvorbu vremenskog EEG signala je GASF ili *Gramian Angular Summation Field*, koji je jedna od dvije varijacije GAF algoritma.



Slika 17 GASF - veličina slike 570*570

Na prethodnoj slici je prikazana jedna od slika dobivenih pretvorbom korištenih vremenskih podataka EEG signala preko GASF algoritma. Veličina slike 570*570, ali to nije veličina koja je korištena u svrhu treniranja i testiranja modela. Za rad s modelom korištena je veličina 100*100 koja je za računalo znatno manje opterećujuća za generirati pri pretvaranju cijelog vremenskog skupa podataka u slike jer je slika znatno manje rezolucije i također lakše se primijete različiti pikseli.



Slika 18 GASF - veličina slike 100*100

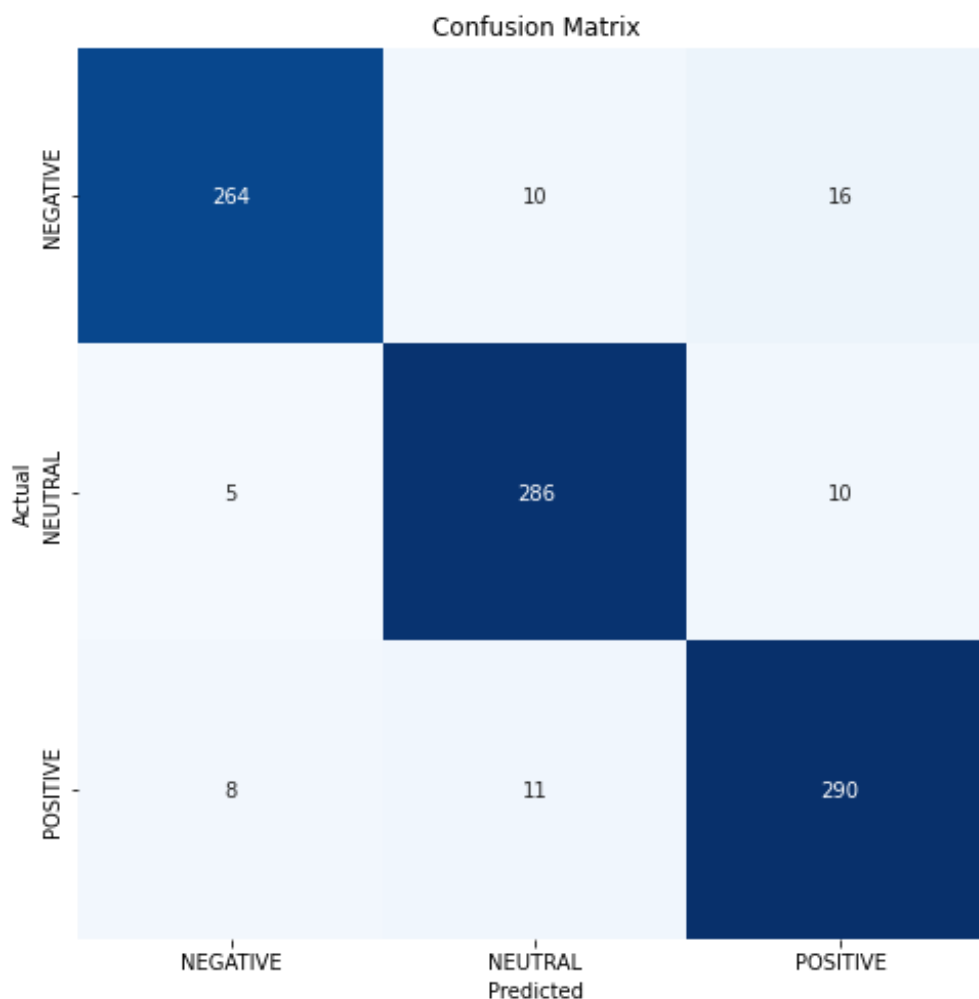
Donja slika pokazuje algoritam korišten za generiranje slika dobivenih iz originalnog skupa podataka koristeći GASF algoritam. Za pretvorbu podataka unutar GAF algoritma nam je potrebno poslati koju veličinu slike želimo generirati te koju od dvije metode želimo koristiti, u ovom slučaju to je metoda zbrajanja što nam GASF varijaciju algoritma. Nakon stvaranja slika koristeći GASF preostalo nam ih je samo još prikazati.

```
image_size = 100
gasf = GAF(image_size, method='summation')
X_gasf = gasf.fit_transform(df.T)

print(X_gasf[10].shape)
plt.figure(figsize=(8, 8))
plt.imshow(X_gasf[10], cmap='viridis', origin='lower')
plt.title('GASF', fontsize=16)
plt.show()
```

Slika 19 GASF - generiranje slika

Kada konačno imamo sve potrebne podatke, generirane slike, možemo ih iskoristiti za treniranje i naravno testiranje modela. Kao i kod treniranja za RNN mrežu korišteno je samo 5 epoha, a istrenirani model na testiranju daje rezultat od 93,333% točnosti.

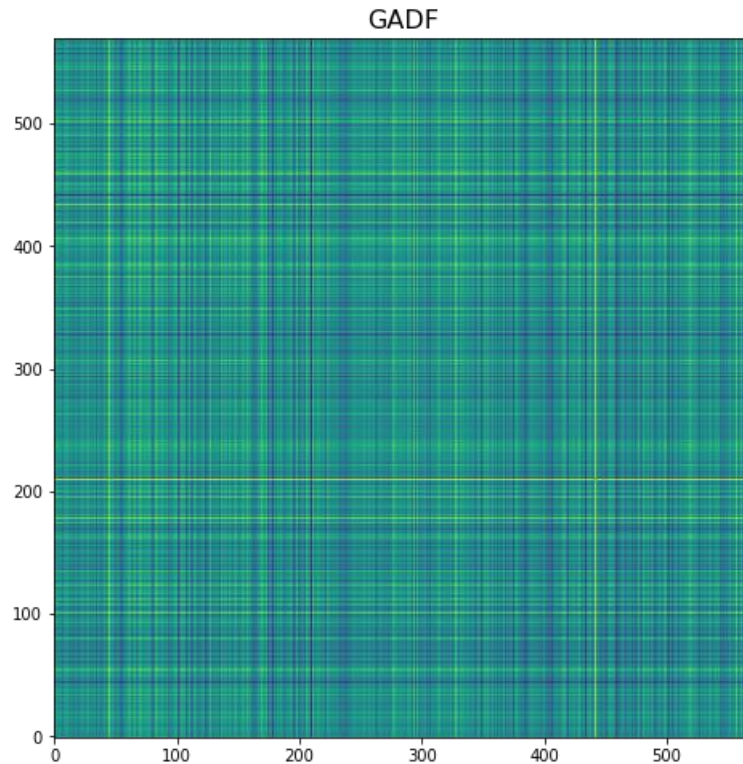


Slika 20 Matrica konfuzije – GASF

Preko prethodno navedenog rezultata točnosti modela već znamo da je točnost u usporedbi s RNN modelom manja, ali preko matrice konfuzije detaljnije možemo vidjeti gdje su te pogreške i nastale. Pojedinačna najveća greška se dogodila u situaciji kada smo imali sliku koja predstavlja negativno stanje, ali naš model je to klasificirao kao da je pozitivno što se dogodilo 16 puta. Također najveća sveukupna greška gledajući prema stanjima nastaje ponovno kada se model susretne s negativnim stanjem. Sveukupan broj greški kada se model susreo s negativnim stanjem je 26, dok najmanju sveukupnu grešku imamo kada se model susreće s neutralnim stanjem za kojeg je napravio pogrešku od 15 puta, a za susretanje s pozitivnim stanjem sveukupna greška je 19.

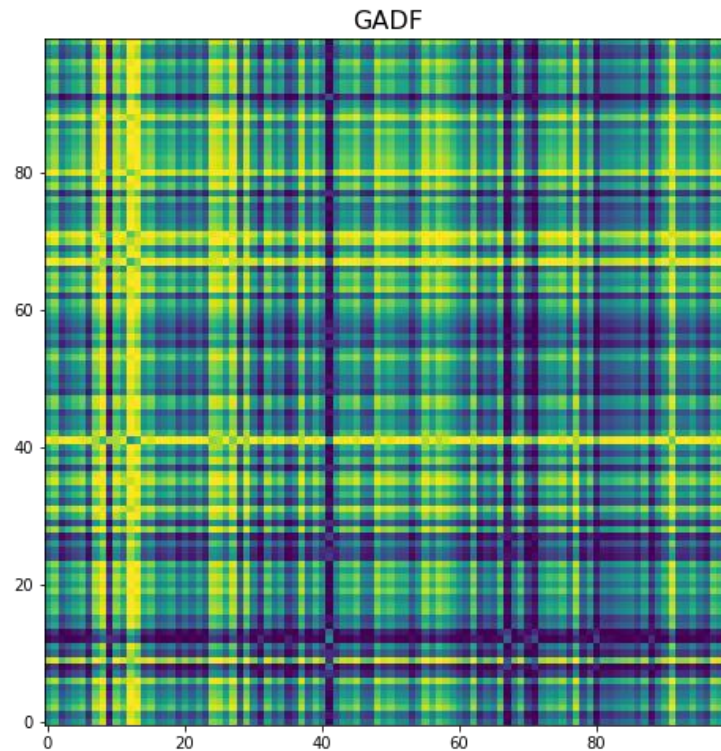
5.4.2. GADF

Za drugi algoritam pretvorbe promatramo GADF ili *Gramian Angular Difference Field*, tj. druga varijacija GAF algoritma.



Slika 21 GADF - veličina slike 570*570

Na prethodnoj slici je prikazana jedna od slika dobivenih pretvorbom korištenih vremenskih podataka EEG signala preko GADF algoritma. Veličina slike je 570*570, ali kao i kod prethodnog slučaja s GASF algoritmom to nije veličina koju koristimo u svrhu treniranja i testiranja modela već je veličina korištene slike 100*100.



Slika 22 GADF - veličina slike 100*100

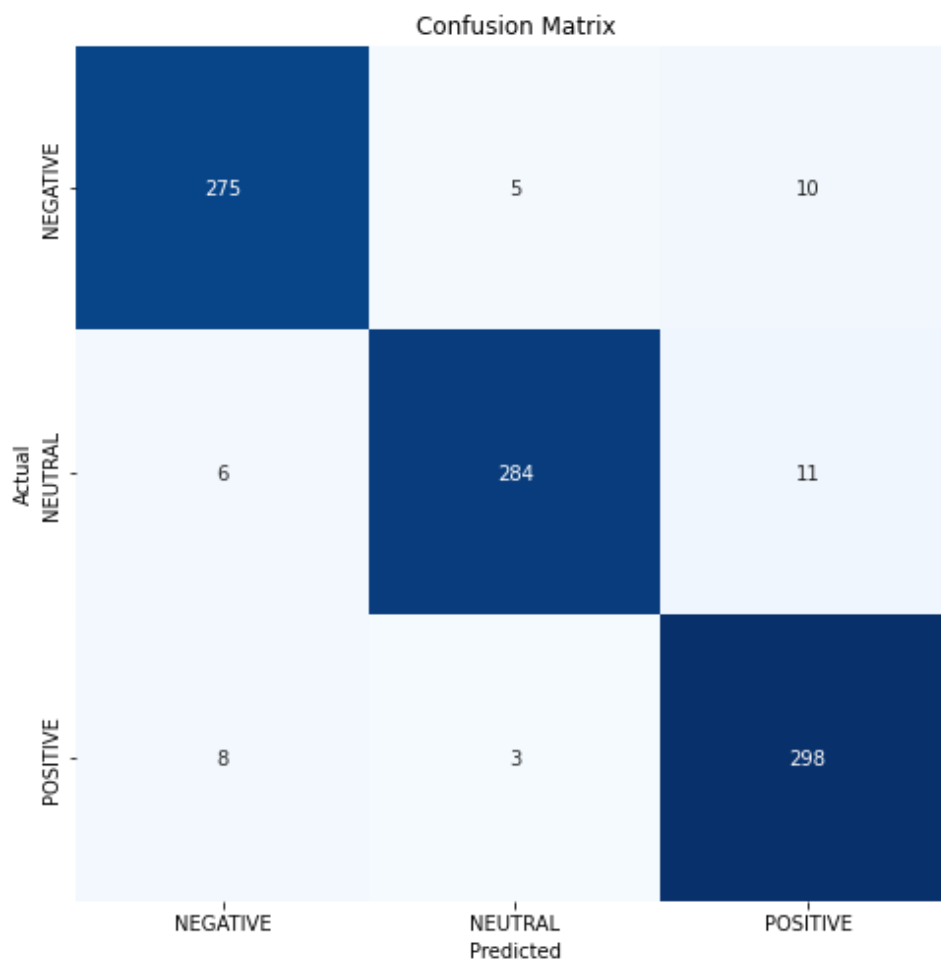
Na donjoj slici je prikazan algoritam korišten za generiranje slika dobivenih iz početnog skupa podataka koristeći GADF algoritam. Kao i kod prethodnog slučaja gdje smo koristili GASF tako su nam i ovdje potrebni isti parametri jer se i dalje radi o GAF algoritmu. Razlika od prethodnog slučaja je u tome koju metodu koristimo gdje smo kod GASF koristili zbrajanje, da bi dobili GADF koristimo razliku.

```
image_size = 100
gadf = GAF(image_size, method='difference')
X_gadf = gadf.fit_transform(df.T)

print(X_gadf[10].shape)
plt.figure(figsize=(8, 8))
plt.imshow(X_gadf[10], cmap='viridis', origin='lower')
plt.title('GADF', fontsize=16)
plt.show()
```

Slika 23 GADF - generiranje slika

Konačno nakon dostupnosti svih podataka pokrećemo proces treniranja modela kroz 5 epoha te pri testiranju kao rezultat dobivamo točnost od 95,222%.

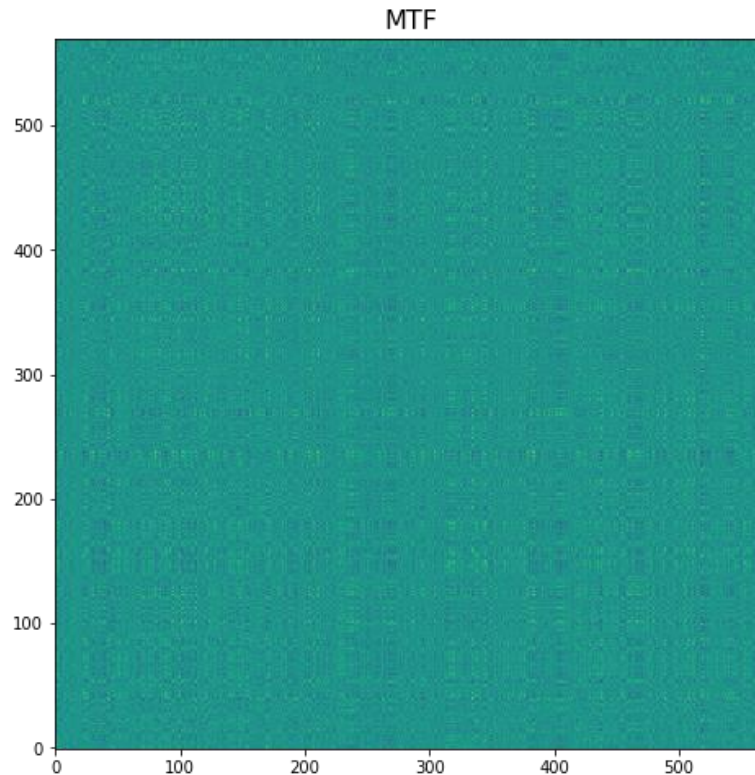


Slika 24 Matrica konfuzije – GADF

Preko dobivenog rezultata za točnost možemo vidjeti da korištenjem GADF algoritma za pretvorbu podataka smo dobili bolji rezultat od onoga kada smo koristili GASF algoritam, ali i dalje razina točnosti je manja od RNN modela gdje smo koristili originalni vremenski skup podataka. Promjenu u rezultatu također možemo vidjeti preko matrice konfuzije, gdje prije smo imali najveću količinu krive klasifikacije kada bi model bio suočen s negativnim stanjem i najmanju pogrešku u slučaju susretanja s neutralnim stanjem, ali sada imamo najveću pogrešku kod neutralnog stanja. Najveća pojedinačna pogreška se pojavila u slučaju kada je model dobio podatke o neutralnom stanju ali ih je zamijenio za pozitivno stanje. Također za sveukupnu najveću pogrešku imamo kod neutralnog stanja gdje je model pogrešno klasifikaciju napravio 17 puta, a najmanja pogreška u ovom slučaju je bila kod klasifikacije pozitivnog stanja gdje imamo pogrešnu klasifikaciju napravljenu 11 puta.

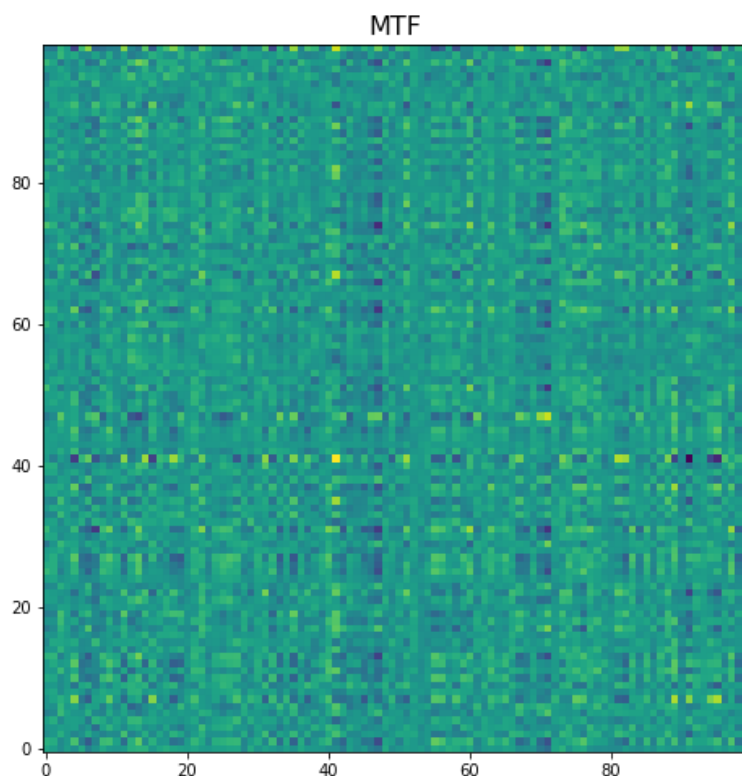
5.4.3. MTF

Treći i posljednji korišteni algoritam za pretvorbu podataka je MTF ili *Markov Transition Field*.



Slika 25 MTF - veličina slike 570*570

Na prethodnoj slici je prikazana jedna od slika koje su dobivene transformacijom originalnog vremenskog EEG skupa podataka veličine 570*570, ali naravno kao i kod prethodna dva algoritma to nije veličina koju ćemo koristiti za treniranje i testiranje modela već veličina koju koristimo je 100*100.



Slika 26 MTF - veličina slike 100*100

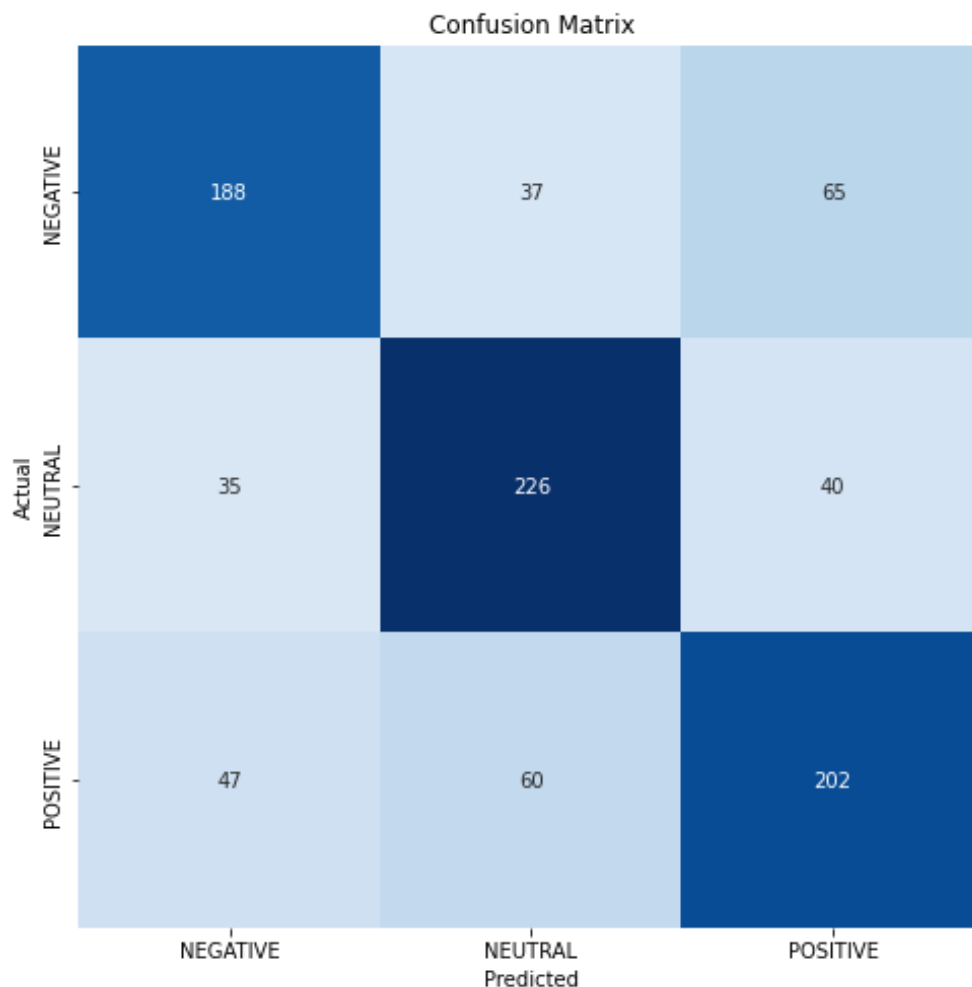
Na donjoj slici je prikazan algoritam korišten za generiranje slika iz početnog vremenskog skupa podataka koristeći MTF algoritam. Možemo vidjeti da parametar koji nam je ponovno potreban je već spomenuta veličina slike, ali za razliku od prethodna dva algoritma ne koristimo više nijedan parametar.

```
image_size = 100
mtf = MTF(image_size)
X_mtf = mtf.fit_transform(df_short)

print(X_mtf[10].shape)
plt.figure(figsize=(8, 8))
plt.imshow(X_mtf[10], cmap='viridis', origin='lower')
plt.title('MTF', fontsize=16)
plt.show()
```

Slika 27 MTF - generiranje slika

Naravno nakon korištenja MTF algoritma i kreiranja potrebnog skupa podataka, koristimo taj skup za treniranje i provjeru modela. Treniranje ponovno izvršavamo kroz 5 epoha, a kao rezultat pri testiranju točnosti dobivamo točnost od 68.444%.



Slika 28 Matrica konfuzije – MTF

Kao što smo već vidjeli preko rezultata točnosti model treniran na skupu podataka dobivenih koristeći MTF algoritam daje znatno najlošiji rezultat od svih prethodnih opcija. To naravno možemo detaljnije i vidjeti na matrici konfuzije gdje imamo znatan broj pogrešaka u klasifikaciji kod sva tri stanja. Pojedinačna najveća pogreška kod klasifikacije u skupu dobivenom korištenjem MTF algoritma je u slučaju pokušaja klasifikacije negativnog stanja gdje model tu sliku klasificira pod pozitivno stanje, a ta pogreška se pojavila 65 puta što je veće od zbroja svih pogrešnih klasifikacija kod prethodnih modela. Sveukupna najveća pogreška koju ovdje imamo je 107 pogrešnih klasifikacija što se pojavilo pri susretanju modela s pozitivnim stanjem, najmanja pogreška klasifikacije je nastala kod neutralnog stanja sa 75 pogrešnih klasifikacija i konačno kod negativnog stanja imamo pogrešku od 102 pogrešne klasifikacije.

5.5. F mjera

Kao dodatni prikaz dobivenih rezultata promatramo rezultate preciznosti, odziva i F mjere. Unutar statističke analize i binarne klasifikacije F mjera (19) je test namijenjen za mjerenje točnosti. Računa se koristeći rezultate preciznosti (17) i odziva (18), gdje je preciznost broj *true positive* rezultata podijeljen s brojem svih pozitivnih rezultata, uključujući i pogrešno identificirane, a opoziv dobivamo tako da dijelimo *true positive* rezultate s brojem svih uzoraka koji su trebali biti identificirani kao pozitivni. Konačno F mjera je težinska harmonijska sredina dobivena od preciznosti i odziva, ali obično se koristi balansirana F1 mjera (20). Najviši rezultat koji možemo dobiti za F mjeru je 1.0, što predstavlja savršenu preciznost i odziv, odnosno 100% točnost modela.

Za primjer od 2 skupa podataka imamo 4 moguća slučaja

- TP – stvarno pozitivni (*true positives*)
- FP – lažno pozitivni (*false positives*)
- FN – lažno negativni (*false negatives*)
- TN – stvarno negativni (*true negatives*)

		Točni entiteti	
		točno	nije točno
Odabrani entiteti	odabrano	TP	FP
	nije odabrano	FN	TN

Tablica 3 Tablica slučaja - 2*2

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (16)$$

$$P = \frac{TP}{TP + FP} \quad (17)$$

$$R = \frac{TP}{TP + FN} \quad (18)$$

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (19)$$

$$\text{za } \beta = 1 \left(\alpha = \frac{1}{2} \right) \quad F1 = \frac{2PR}{P + R} \quad (20)$$

RNN	preciznost	odziv	F1
Negative	0,97	0,95	0,96
Neutral	1,00	0,99	0,99
Positive	0,94	0,978	0,96
Acc			0,97

Tablica 4 F mjera - RNN

CNN – GASF	preciznost	odziv	F1
Negative	0,95	0,91	0,93
Neutral	0,93	0,95	0,94
Positive	0,92	0,94	0,93
Acc			0,93

Tablica 5 F mjera - CNN (GASF)

CNN – GADF	preciznost	odziv	F1
Negative	0,95	0,95	0,95
Neutral	0,97	0,94	0,96
Positive	0,93	0,96	0,95
Acc			0,95

Tablica 6 F mjera - CNN (GADF)

CNN – MTF	preciznost	odziv	F1
Negative	0,70	0,65	0,67
Neutral	0,70	0,75	0,72
Positive	0,66	0,65	0,66
Acc			0,68

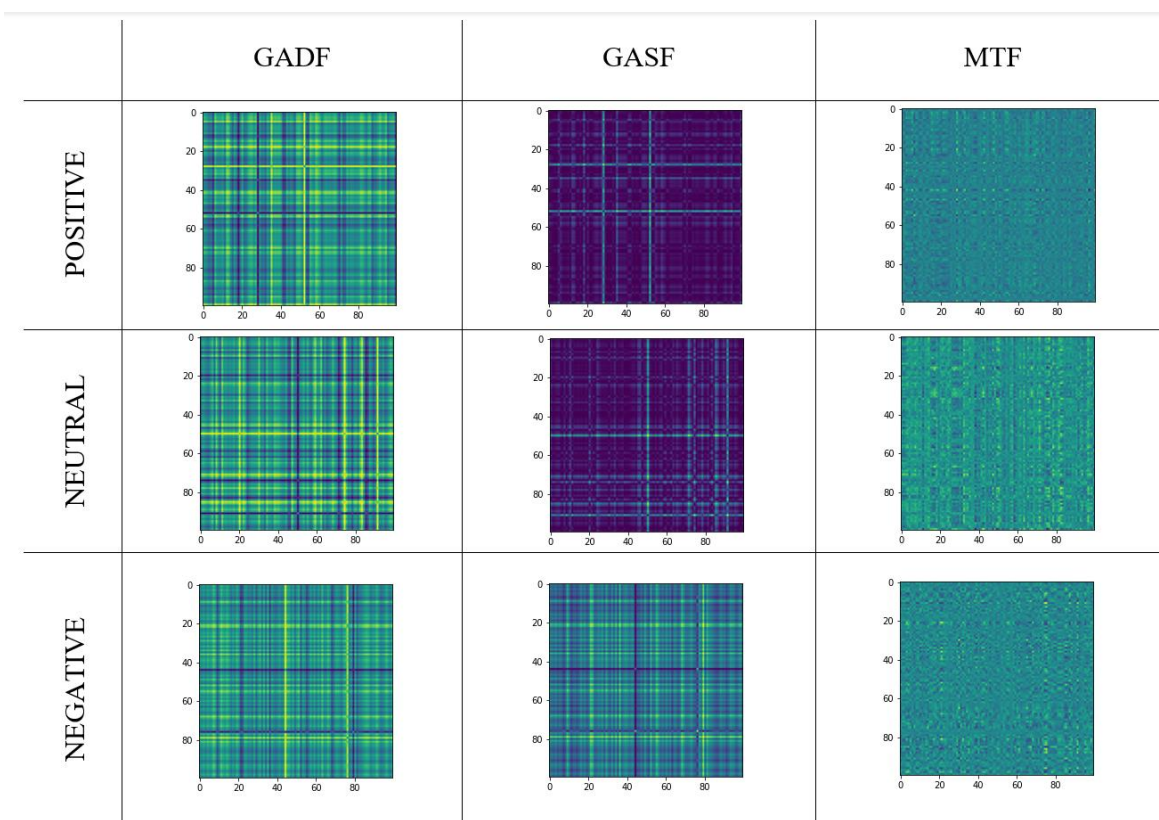
Tablica 7 F mjera - CNN (MTF)

5.6. Konačna opažanja

Kao što se već možemo vidjeti iako korištenjem kovolucijskih neuronskih mreža uspjeli smo dobiti jako dobre rezultate, korištenjem rekurentne neuronske mreže smo ipak dobili nešto bolji rezultat. Kod korištenja konvolucijske neuronske mreže rezultati točnosti koji su se istakli u pozitivnom smislu su dobiveni koristeći skup podataka generiran korištenjem GAF varijacija algoritma. Najbolji dobiveni rezultat točnosti pri korištenju konvolucijske neuronske mreže je dobiven slike generirane GADF algoritmom, koji je 95.222% dok je najveća dobivena F1 mjera jednaka 0,96 i dobivena je za neutralno mentalno stanje. GASF algoritam iako s lošijim rezultatom od svoje druge varijante, dobio je visoku razinu točnosti od 93,333% te također kao i kod GADF algoritma najveća dobivena F1 mjera je za neutralno stanje a jednaka je 0,94. Pored dobrih rezultata točnosti dobivenih koristeći se slikama generiranih preko GAF algoritma, također smo dobili i jedan lošiji rezultat koristeći slike dobivene preko MTF algoritma. Točnost koju smo dobili

kod MTF algoritma je 68,444%, a najveća F1 mjera ponovno kod neutralnog mentalnog stanja je 0,72.

Naravno rezultat dobiven koristeći se slikama generiranih MTF algoritmom nije usporediv s točnosti dobivenom koristeći se originalnim vremenskim skupom podataka koji smo koristili za rad s rekurentnom neuronskom mrežom koji je 97,031%, a najbolja F1 mjera 0,99 kod neutralnog mentalnog stanja. Iako pri usporedbi s rezultatima dobivenim koristeći MTF je znatno primjetna, pri usporedbi s rezultatima dobivenim koristeći GADF nije toliko značajna. Najmanja razlika koju smo uspjeli postići koristeći konvolucijsku neuronsku mrežu je 1,809%.



Slika 29 GADF, GASF, MTF – usporedba

Na slici poviše imamo vizualni prikaz korištenih skupova podataka. Točnije rečeno imamo prikazanu sliku za svako od tri korištena mentalna stanja kod svakog od korištenih algoritama za njihovo generiranje. Iako ne možemo uzeti svaku od prikazanih slika kao reprezentaciju čitavog skupa, možemo ih promatrati u svrhu osnovne vizualne analize.

Na prvo promatranje možda možemo reći da ljudskom oku sve slike većinski izgledaju jednako ili jako slično. Iako nisu u potpunosti jednake naravno dobivene slike su dobivene iz sličnih vremenskih podataka te je za očekivati da će i njihovi prikazi biti primjetno slični.

Pri boljem promatranju možemo vidjeti da najveće razlike unutar pojedinačnih slika i među slikama možemo vidjeti kod slika dobivenih pri korištenju GADF algoritma. Što je možda i za očekivati s obzirom da smo pri korištenju slika generiranih GADF algoritmom dobili najveće rezultate točnosti. Dok kod slika koje imamo dobivene korištenjem GASF algoritma imamo sličnu situaciju kao i kod slika dobivenih GADF algoritmom, što je i za očekivati jer se radi o varijacijama istog algoritma i također što objašnjava slične rezultate točnosti, kod korištenja MTF algoritma imamo vidnu razliku kod generiranih slika. Za slike dobivene GAF varijacijama algoritma imamo primjetne razlike boja unutar pojedinačnih slika te vidljive razlike među slikama, za MTF algoritam imamo veliku količinu jednoličnosti. Ne pojavljuju se primjetne razlike u bojama, te i same razlike među slikama različitih mentalnih stanja su manje primjetne. Za ljudsko oko razlikovati ovakve slike je jako teško, ali iako je računalu to znatno lakše i dalje pri klasifikaciji može dovesti do poteškoća. Jednoličnost među dobivenim slikama od strane MTF algoritma može objasniti razlog značajne razlike među razinama točnosti pri treniranju modela konvolucijske neuronske mreže.

Zaključak

Nakon provedene analize korištenih podataka i načina njihove obrade jesmo li u mogućnosti odgovoriti na početno pitanje? Ima li smisla koristiti konvolucijske neuronske mreže za analizu vremenski podataka, odnosno pretvaranje vremenskih podataka u slike koje su pogodne za korištenje kod konvolucisjkih neuronskih mreža?

Prije samog odgovora na ovo pitanje trebamo se podsjetiti na problem koji je nastao kod transformacije i korištenja transformiranih podataka. Taj problem je ograničenost korištenog hardvera koji nije mogao podnijeti kompletnu količinu korištenih podataka te smo bili ograničeni na korištenje samo jedne manje količine generiranih slika u svrhu treniranja te kasnije testiranja korištenih CNN modela, dok smo za početni RNN model bili u mogućnosti koristiti cijeli početni vremenski skup podataka.

Usprkos pojavi tehničkih problema možemo reći da smo dobili pretežno zadovoljavajuće rezultate. Modeli trenirani na slikama koje smo dobili koristeći GAF algoritme su imali jako dobre rezultate točnosti uz samo jedan lošiji rezultat točnosti koji je postignut koristeći slike dobivene preko MTF algoritma. Iako dobiveni rezultat točnosti kod sva tri slučaja korištenja konvolucijske neuronske mreže je manji od onoga kod korištenja rekurentne neuronske mreže sa originalnim vremenskim podacima, dobivene rezultate ne trebamo odbaciti. Najmanja razlika u točnosti je dobivena pri korištenju GADF algoritma, ali ipak taj model je treniran na znatno manjem skupu podataka od originalnog modela.

Zaključak je da korištenje CNN modela i pretvorba vremenskih podataka u slike nije ideja koju treba odbaciti, ali istraživanje bi se također trebalo ponovno provesti s uklonjenim tehničkim ograničenjima da možemo dobiti točnije rezultate.

Literatura

- [1] Chao-Lung Yang, Zhi-Xuan Chen, Chen-Yi Yang, >>Sensor Classification Using Convolutional Neural Network by Encoding Multivariate Time Series as Two-Dimensional Colored Images<<, 2019
- [2] Nihal Fatma Gu"ler, Elif Derya U" beyli, I"nan Gu"ler, >> Recurrent neural networks employing Lyapunov exponents for EEG signals classification<<, 2005
- [3] Wan Woaswi, Malik Hanif, S. B. Mohamed, Noraini Hamzah, Zairi Ismael Rizman, >> Human Emotion Detection Via Brain Waves Study by Using Electroencephalogram (EEG)<<, International Journal on Advanced Science Engineering and Information Technology, 2016
- [4] J. J. Bird, L. J. Manso, E. P. Ribiero, A. Ekart, and D. R. Faria, >>A study on mental state classification using eeg-based brain-machine interface<<,in 9th International Conference on Intelligent Systems, IEEE, 2018
- [5] G'abor Petneh'azi, >>Recurrent Neural Networks for Time Series Forecasting<<, 2019
- [6] Kyung-OkCho, Hyun-Jong Jang, >>Comparison of diferent input modalities and network structures for deep learning-based seizure detection<<, Scientific reports, 2020
- [7] Catarina da Silva Lourenço, Marleen C. Tjepkema-Cloostermans, Michel J.A.M. van Putten, >>Efficient use of clinical EEG data for deep learning in epilepsy<<, 2021
- [8] Keiron O'Shea, Ryan Nash, >>An Introduction to Convolutional Neural Networks<<, 2015
- [9] Jayanth Koushik, >>Understanding Convolutional Neural Networks<<, 29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain
- [10] M. Teplan, >>Fundamentals of EEG Measurement<<, MEASUREMENT SCIENCE REVIEW, Volume 2, Section 2, 2002
- [11] Priyanka A. Abhang, Bharti W. Gawali, Suresh C. Mehrotra, >>Chapter 2 - Technological Basics of EEG Recording and Operation of Apparatus<<, 2016

- [12] David Petersson, >>CNN vs. RNN: How are they different?<<, TeachTarget, 2021, <https://www.techtargget.com/searchenterpriseai/feature/CNN-vs-RNN-How-they-differ-and-where-they-overlap>
- [13] Stephen Grossberg, Scholarpedia, >>Recurent Neural Networks<<, 2013 [doi:10.4249/scholarpedia.1888](https://doi.org/10.4249/scholarpedia.1888)
- [14] Z. Wang, T. Oates, >>Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks.<<, AAAI Workshop, 2015

Ilustracije

Slika 1 Muse headband.....	2
Slika 2 Standardne pozicije elektroda	3
Slika 3 Primjer EEG signala s Muse senzora	3
Slika 4 Primjer moždanih valova	7
Slika 5 Očitavanje napadaja preko EEG-a	8
Slika 6 Slikoviti prikaz RNN arhitekture	10
Slika 7 LSTM osnovna arhitektura.....	11
Slika 8 GRU osnovna arhitektura.....	12
Slika 9 Slikoviti prikaz CNN arhitekture	13
Slika 10 Vizualizacija kovolucijskog sloja.....	14
Slika 11 Vizualizacija Max Pool (2*2).....	15
Slika 12 GASF, GADF, MTF rezultat pretvorbe	19
Slika 13 Odsječak vremenskih podataka	21
Slika 14 RNN model	22
Slika 15 Matrica konfuzije – RNN	23
Slika 16 CNN model	24
Slika 17 GASF - veličina slike 570*570	25
Slika 18 GASF - veličina slike 100*100	26
Slika 19 GASF - generiranje slika.....	26
Slika 20 Matrica konfuzije – GASF	27
Slika 21 GADF - veličina slike 570*570.....	28
Slika 22 GADF - veličina slike 100*100.....	29
Slika 23 GADF - generiranje slika	29

Slika 24 Matrica konfuzije – GADF.....	30
Slika 25 MTF - veličina slike 570*570	31
Slika 26 MTF - veličina slike 100*100	32
Slika 27 MTF - generiranje slika.....	32
Slika 28 Matrica konfuzije – MTF	33
Slika 29 GADF, GASF, MTF – usporedba	37

Tablice

Tablica 1 Podatci o moždanim valovima	6
Tablica 2 CNN i RNN razlike	16
Tablica 3 Tablica slučaja - 2*2.....	34
Tablica 4 F mjera - RNN	35
Tablica 5 F mjera - CNN (GASF)	35
Tablica 6 F mjera - CNN (GADF).....	36
Tablica 7 F mjera - CNN (MTF)	36