SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1158

# ARHITEKTURE ZA RASPOZNAVANJE SLIKA SLOJEVIMA PAŽNJE

Darijan Gudelj

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 1158

# ARHITEKTURE ZA RASPOZNAVANJE SLIKA SLOJEVIMA PAŽNJE

Darijan Gudelj

Zagreb, lipanj 2023.

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 10. ožujka 2023.

# ZAVRŠNI ZADATAK br. 1158

| | |
|---|---|
| Pristupnik: | **Darijan Gudelj (0036529361)** |
| Studij: | Elektrotehnika i informacijska tehnologija i Računarstvo |
| Modul: | Računarstvo |
| Mentor: | prof. dr. sc. Siniša Šegvić |

Zadatak: **Arhitekture za raspoznavanje slika slojevima pažnje**

Opis zadatka:

Raspoznavanje slika važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Trenutno stanje tehnike koristi duboke modele koji se uče s kraja na kraj. U posljednjih nekoliko godina posebno su zanimljivi modeli koji umjesto konvolucije koriste slojeve pažnje. Ovaj rad proučava pristupe za njihovo organiziranje u funkcionalne cjeline. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće pristupe za klasifikaciju slike. Odabrati slobodno dostupni skup slika te oblikovati podskupove za učenje, validaciju i testiranje. Predložiti prikladnu arhitekturu dubokog klasifikacijskog modela. Uhodati postupke učenja modela i validiranja hiperparametara. Primijeniti naučene modele te prikazati i ocijeniti ostvarene rezultate. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

# Sažetak

# 1. Uvod

Automatsko prepoznavanje slika može omogućiti značajne ekonomske dobitke i olakšati život ljudima u raznim područjima. Postoje brojne primjene, u medicinskoj slikovnoj dijagnostici, autonomnim vozilima, robotici i automatizaciji, kontrolama kvalitete te proširenoj stvarnosti.

U posljednjem desetljeću, transformeri su postali dominatna arhitektura za obradu prirodnog jezika, ostvarujući izvanredne rezultate u različitim zadacima poput strojnog prevođenja i jezičnog modeliranja[1]. Međutim, primjena transformera na vizualne podatke, kao što su slike ili videozapisi, predstavlja izazov zbog kvadratne kompleknosti pažnje. U proteklim godinama, vizualni transformer[2] postao je standardni model koji se koristi za rješavanje tih problema.

Jednom izračunata značajka vizualnog transformera mora proći kroz sve sljedeće slojeve, što smanjuje kapacitet modela. Model mora odlučiti želi li zadržati prethodno izračunate značajke ili dodati nove. Ovaj problem može se riješiti povećanjem broja parametara modela, ali to rezultira većom potrebom za memorijom i računalnim resursima te povećava latenciju predikcija.

U ovom radu istražujemo načine na koje možemo povezati slojeve vizualnog transformera kako bismo povećali ponovno korištenje značajki i poboljšali parametarsku efikasnost modela. U tu svrhu proučavamo četiri različite arhitektonske modifikacije baznog modela.

U konvolucijskim modelima, problemi ponovne uporabe značajki riješeni su DenseNet arhitekturom[3], no takva rješenja još nisu pokazana u transformerima.

Radovi koji se bave smanjenjem komputacije tokena kao što su spajanje tokena[4] i prestanak komputacije[5] iako slični ortogonalni su na ovaj rad.

# 2. Klasifikacija slika

Klasifikacija slika izazovan je zadatak u računalnom vidu zbog visoke varijabilnosti i složenosti prirodnih slika. Cilj klasifikacije slika je dodijeliti jednu ili više oznaka ulaznoj slici, ukazujući na prisutnost ili odsutnost određenih objekata, uzoraka ili atributa na slici. Ovaj zadatak ima mnoge primjene, od pretraživanja slika temeljenog na sadržaju do autonomne vožnje, gdje su precizno i brzo prepoznavanje vizualnih znakova ključni za donošenje odluka.

## 2.1. Arhitekture za klasifikaciju slika

Uspjeh u klasifikaciji slika uvelike ovisi o izboru temeljne arhitekture. U posljednjim godinama razvijeno je nekoliko arhitektura koje su pokazale izvanredne rezultate na različitim zadacima klasifikacije. Te arhitekture služe kao moduli većeg rješenja ili su samostalne i treniraju se od početka do kraja.

### 2.1.1. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (CNN) postale su popularna klasa dubokih neuronskih mreža za klasifikaciju slika zbog svoje izvrsne performanse u usporedbi s tradicionalnim metodama strojnog učenja. CNN koristi niz konvolucijskih slojeva koji izvode lokalne linearno-transformacijske operacije nad ulaznim slikama kako bi izvukli relevantne značajke, istovremeno očuvajući prostorne odnose među njima. Ti konvolucijski slojevi slijede nelinearne aktivacijske funkcije, poput funkcije ReLU (Rectified Linear Unit), koje poboljšavaju diskriminacijsku moć mreže.

Izlaz zadnjeg konvolucijskog sloja je skup značajkovnih mapa koje se zatim ravna i prolazi kroz potpuno povezane slojeve, poznate i kao glava za klasifikaciju, kako bi se generirale konačne predikcije. Tijekom faze treniranja, težine mreže se ažuriraju kroz propagaciju unatrag (backpropagation), pri čemu se izračunavaju gradijenti gubitka funkcije u odnosu na težine i koriste za ažuriranje težina.

U literaturi su predložene različite arhitekture konvolucijskih neuronskih mreža (CNN), pri čemu svaka ima svoje jedinstvene dizajnerske karakteristike. AlexNet[6] je bio pobjednički model na natjecanju ImageNet Large scale Visual Recognition Challenge 2012[7]. Koristio je ReLU aktivacijsku funkciju, dropout regularizaciju i tehnike povećanja podataka. S druge strane, VGGNet[8] je povećao dubinu mreže dodavanjem više slojeva konvolucije i manjih veličina filtara. InceptionNet[9] je predložio korištenje Inception modula koji izvode paralelne konvolucije s različitim veličinama jezgara kako bi uhvatili višeskalne značajke. ResNet[10], još jedna popularna arhitektura, uvela je rezidualne veze koje omogućuju bolje rješavanje nestajućih gradijenata tijekom treniranja. Nedavno je predložen EfficientNet[11], koji optimizira dubinu, širinu i rezoluciju mreže koristeći metodu kombiniranog skaliranja, rezultirajući boljim performansama i smanjenom računalnom složenošću.

DenseNet je još jedna popularna arhitektura CNN-a koja je privukla pažnju u posljednjim godinama. DenseNet uvodi jedinstven uzorak povezivanja između slojeva, pri čemu svaki sloj prima ulaz iz svih prethodnih slojeva, što rezultira izrazito gustom povezanošću između slojeva. Ovaj uzorak povezivanja potiče ponovnu upotrebu značajki i omogućuje mreži da nauči kompaktnije reprezentacije, istovremeno smanjujući broj parametara. DenseNet je pokazao najnovije rezultate na raznim zadacima klasifikacije slika, zahtijevajući manje parametara i računskih operacija u usporedbi s drugim arhitekturama CNN-a. Osim toga, DenseNet je proširen na druge zadatke računalnog vida, poput detekcije objekata i semantičke segmentacije, gdje je također postigao obećavajuće rezultate.

Iako popularani u literaturi, ova rad ne proučava osnovne CNN-ove detaljnije.

### 2.1.2. Modeli s pažnjom

Mehanizmi pažnje su posljednjih godina dobili značajan interes kao efektivna tehnika za poboljšanje performansi dubokih neuronskih mreža, uključujući one koje se koriste za klasifikaciju slika. Osnovna ideja iza pažnje je omogućiti mreži da selektivno usmjeri komputaciju na najinformativnije dijelove ulaza, što može dovesti do preciznijih predviđanja i smanjene računarske složenosti. U posljednjih nekoliko

godina predloženi su razni modeli temeljeni na pažnji, uključujući Vision Transformer (ViT), Swin Transformer[12] i druge varijante transformera proširenih slikama.

ViT je model baziran na transformeru koji zamjenjuje tradicionalne konvolucijske slojeve u konvolucijskim neuronskim mrežama (CNN) slojevima samopažnje, što omogućuje efikasno izdvajanje značajki i globalno razumijevanje slika. Model uči usmjeravati pažnju na različite dijelove slike i agregira značajke na koje je usmjerena kako bi klasificirao sliku. Povećane verzije ViT-a postigle su vrhunske rezultate na nekoliko referentnih skupova podataka, poput ImageNet-a i COCO-a.

Swin Transformer još je jedan model baziran na transformeru koji uvodi hijerarhijsku arhitekturu tako da dijeli ulaznu sliku na manje dijelove i obrađuje ih transformerskim slojevima na različitim razinama. Ovaj pristup poboljšava sposobnost modela da uhvati lokalne i globalne kontekstualne informacije i pokazao je izvanredne performanse u raznim zadacima klasifikacije slika.

## 2.2. Treniranje

U ovom odjeljku raspravljamo o različitim metodama obuke klasifikatora slika, uključujući krajnje nadziranu obuku, fino podešavanje prethodno obučenih modela, kontrastnu obuku teksta i slike te jezične modele s ugrađenim slikama.

### 2.2.1. Nadzirano učenje

Najjednostavniji pristup obuci klasifikatora slika uključuje korištenje modela poput konvolucijskih neuronskih mreža ili vizualnih transformatora s glavom za klasifikaciju koja je obučena optimizirati gubitak križne entropije. Model se obučava na dovoljno velikom skupu podataka, poput ImageNet-a, kako bi naučio temeljne značajke slika.

Tijekom obuke, mreži se daju označene slike, a izračunava se gubitak križne entropije između predviđenih vjerojatnosti razreda i stvarnih oznaka. Gradijenti gubitka se zatim propagiraju unatrag kroz mrežu, a parametri modela se ažuriraju koristeći optimizator poput stohastičkog gradijentnog spusta (SGD) ili Adam-a[13].

Ovaj pristup je jednostavan i učinkovit, te se koristi za obuku najnaprednijih klasifikatora slika poput ResNet-a, EfficientNet-a i ViT-a. Međutim, za postizanje dobre performanse zahtijeva veliku količinu označenih podataka.

### 2.2.2. Fino podešavanje prethodno naučenih modela

Prilagodba unaprijed naučenih modela je uobičajeni pristup u dubokom učenju, gdje se unaprijed naučeni model dalje trenira na ciljnom zadatku koristeći manji skup podataka. Postupak unaprijednog treniranja uključuje treniranje modela na velikom skupu podataka koristeći nadzirani cilj, poput gubitka križne entropije, ili polu-nadziranog cilja, poput kontrastnog učenja ili rekonstrukcijskog učenja. Kontrastno učenje postalo je popularno u posljednjim godinama jer omogućuje učinkovito treniranje dubokih neuronskih mreža koristeći podatke bez oznaka. Primjeri popularnih metoda kontrastnog učenja uključuju Momentum Contrast (MoCo)[14] i Bootstrap Your Own Latent (BYOL)[15]. Maskirani autoenkoder (MAE)[16] je još jedan popularan pristup gdje mreža uči "popuniti" nedostajuće dijelove slike.

Kada se dobije unaprijed naučeni model, može se prilagoditi na ciljni zadatak, poput klasifikacije slika ili otkrivanja objekata, prilagođavanjem njegovih težina koristeći nadzirano učenje s manjim skupom označenih podataka. Prilagodba unaprijed naučenog modela može značajno smanjiti vrijeme treniranja i poboljšati performanse modela na ciljnom zadatku.

### 2.2.3. Kontrastivno učenje teksta i slika

Tekst-slika kontrastno obučavanje je nedavni pristup koji je pokazao iznimne rezultate na nekoliko referentnih skupova za klasifikaciju slika bez oznaka. Glavna ideja ovog pristupa je obučiti model da ugradi slike i njihove odgovarajuće oznake u zajednički prostor ugradnje, gdje se maksimizira sličnost između slike i njezine oznake, dok se minimizira sličnost između slike i neodgovarajuće oznake.

Postupak obuke uključuje dvije glavne korake: kodiranje slike i teksta te kontrastno učenje. Tijekom kodiranja slike i teksta, slike i njihove odgovarajuće oznake kodiraju se u vektore značajki pomoću CNN-a ili ViT-a za slike te transformatorskog jezičnog modela, poput GPT-a ili BERT-a [17], za oznake.

Tijekom kontrastnog učenja, model je obučen da maksimizira sličnost između ugradnji svake slike i njene odgovarajuće oznake, istovremeno minimizirajući sličnost između ugradnji iste slike i neodgovarajuće oznake. To se postiže korištenjem funkcije gubitka kontrasta, poput trostrukog gubitka ili gubitka InfoNCE.

Ovaj pristup ne zahtijeva oznake na ciljnom skupu podataka, budući da se model obučava na kombinaciji označenih i neoznačenih podataka.

### 2.2.4. Ugrađivanje slikovnih reprezentacija u jezične modele

Jezični modeli, poput GPT-a i BERT-a, pokazali su se učinkovitima u hvatanju semantike teksta i koriste se u nekoliko zadataka obrade prirodnog jezika. Međutim, ovi modeli nisu dizajnirani za obradu vizualnih informacija, niti mogu obraditi vizualne informacije.

Kako bi se prevladala ova ograničenost, nedavni radovi su predložili proširenje jezičnih modela s ugrađenim slikama te različite načine uzemljavanja jezičnih modela u vizualnu domenu. Ideja je zamijeniti jedan ili više tokena u ulaznom nizu s ugrađenim prikazom slike te prilagoditi ugrađujući prostor tome jezičnog modela.

Tijekom finog podešavanja, modelu se daje kombinacija teksta i slika, a cilj je predvidjeti neke zasebne ili tokene u nizu, dajući druge tokene i ugrađenu sliku.

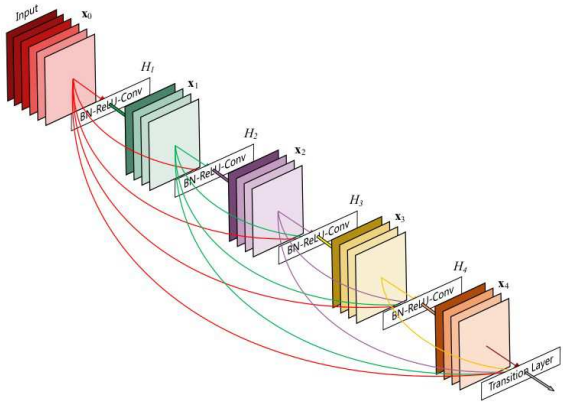
Ovaj pristup omogućuje modelu da uhvati i semantičke i vizualne informacije ulaznog niza, što dovodi do bolje izvedbe na zadacima koji zahtijevaju razumijevanje i teksta i slika, poput vizualnog odgovaranja na pitanja i opisivanja slika.

Neki radovi koji koriste ovu tehniku uključuju Deepmindov Flamingo[18], Googleov PALM-E[19] i VisualBERT[20]. OpenAI GPT4[21] također ima varijantu transformera koja prihvaća slike kao ulaz.

Klasifikacija u ovoj paradigmi se svodi na postavljanje pitanja modelu u prirodnom jeziku i s priloženom slikom i opcijama za odgovor.

# 3. Modeli s gusto povezanim jedinicama

DenseNet je arhitektura dubokog učenja koja je stekla popularnost zbog svoje izuzetne izvedbe u različitim zadacima računalnog vida. Predstavili su je Huang i suradnici 2017. godine. DenseNet je konvolucijska neuronska mreža koja rješava problem nestajućeg gradijenta povezivanjem svakog sloja sa svakim sljedećim slojem na način usmjerenog prosljeđivanja. Ova arhitektura je pokazala izvrsne rezultate na nekoliko referentnih skupova podataka, uključujući ImageNet i CIFAR. DenseNet arhitektura se temelji na ideji gustih veza, gdje je svaki sloj povezan sa



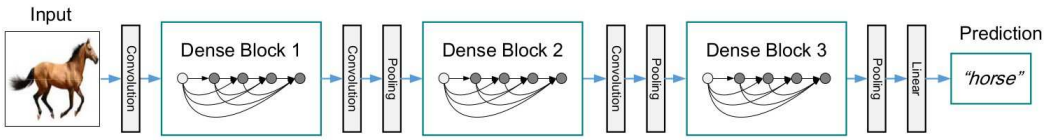

Slika 1: Slika je uzeta iz [3]. Slika prikazuje jedan DenseNetov gusti blok. Prikazane su konekcije između komponenata bloka.

svakim sljedećim slojem u mreži. Ulaz u mrežu prolazi kroz konvolucijski sloj, koji je zatim praćen slojem normalizacije grupiranja i aktivacijom ReLU (rectified linear unit). Izlaz tog sloja se zatim konkatenira sa ulazom svakog sljedećeg sloja, formirajući gusto blok. Svaki gusto blok se sastoji od više konvolucijskih slojeva, slojeva normalizacije grupiranja i aktivacija ReLU. Konačni gusto blok je povezan sa slojem globalnog prosječnog grupiranja, a zatim slijedi sloj softmax koji proizvodi konačni izlaz mreže.

Guste veze u DenseNet omogućuju gradijentima da se lakše prenose kroz mrežu, što smanjuje problem nestajućeg gradijenta koji se javlja u tradicionalnim mrežama usmjerenog prosljeđivanja. Dodatno, DenseNet smanjuje broj parametara dijeljenjem značajnih mapa između slojeva, što čini mrežu učinkovitijom u korištenju memorije i omogućuje obuku na manjim skupovima podataka.

Jedna od ključnih prednosti DenseNet arhitekture u odnosu na druge arhitekture je sposobnost korištenja informacija iz ranijih slojeva, što omogućuje učenje složenijih značajki i poboljšanje izvedbe.

U ovom radu koristimo DenseNet arhitekturu kao inspiraciju za poboljšanje ponovne upotrebe značajki u vizualnim transformerima.

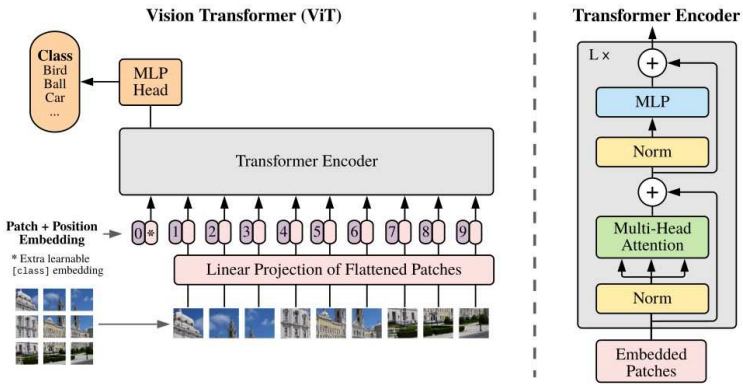



Slika 2: Slika je uzeta iz [3]. Slika prikazuje cijelu DenseNet arhitekturu. Sastoji se od nekoliko gustih blokova koji su spojeni s poolingom i konvolucijama.

# 4. Modeli sa slojevima pažnje

Vision Transformer (ViT) arhitektura je neuronske mreže koja je razvijena za računalni vid. Predstavljena je 2020. godine i temelji se na Transformer arhitekturi koja je izvorno predložena za obradu prirodnog jezika. ViT je model sa samo pažnjom koji obrađuje slike kao sekvence segmenta umjesto pojedinačnih piksela i postigao je rezultate najviše kvalitete na nekoliko referentnih skupova podataka.

Osnovna ideja iza ViT-a je koristiti mehanizam samo pažnje za hvatanje odnosa između različitih dijelova slike. Kod tradicionalnih konvolucijskih neuronskih mreža, ulazna slika se obrađuje putem niza konvolucijskih i združenih slojeva koji postupno smanjuju prostornu rezoluciju i povećavaju apstraktnost. Nasuprot tome, ViT djeluje na sliku punom rezolucijom tako da je dijeli na segmente koji se ne preklapaju, pri čemu se svaki segment tretira kao zaseban token. Ti segmenti se zatim ravnaju u sekvencu i prolaze kroz niz Transformer blokova, koji računaju skup težina pažnje za svaki segment na temelju njegove sličnosti s drugim segmentima u sekvenci i transformiraju tokene. Poseban 'CLS' token se koristi za donošenje konačne predikcije, prolazi kroz mali višeslojni perceptron (MLP) s tangens hiperbolnom funkcijom kao nelinearnošću u jednom skrivenom sloju.




Slika 3: Ukupna arhitektura Vision Transformera. Slika se pretvara u segmente koji se prosljeđuju transformeru. Izlaz 'CLS' tokena se prosljeđuje kroz MLP kako bi se dobio konačni izlaz. Slika preuzeta iz [2].

Jedna od ključnih prednosti ViT-a je da je skalabilan s obzirom na uloženu količinu komputacija. ViT je pokazao visoku modularnost i fleksibilnost, što mu omogućuje prilagodbu širokom rasponu zadataka prepoznavanja slika uz minimalne promjene u arhitekturi. Transformerska pozadina ViT-a omogućuje lagani prijenos znanja i najnovijih tehnika iz obrade prirodnog jezika i time omogućuje ubrzani razvoj.

Međutim, ViT također ima neka ograničenja. Jedan od glavnih izazova je potreba za velikim brojem primjera za treniranje kako bi se postigla visoka preciznost, zbog nedostatka prethodnih informacija i izazova u optimizaciji. CNN-ovi imaju pretpostavku o lokalnosti, dok transformeri nemaju ništa slično tome.

# 5. Metoda

Koristili smo nekoliko alata u ovom radu. Koristili smo PyTorch[22] kao biblioteku za automatsku diferencijaciju, Cifar-10[23] kao skup podataka, AdamW[24] kao optimizator, kosinusni planer stope učenja s linearnim zagrijavanjem, glađenje oznaka i propadanje težina.

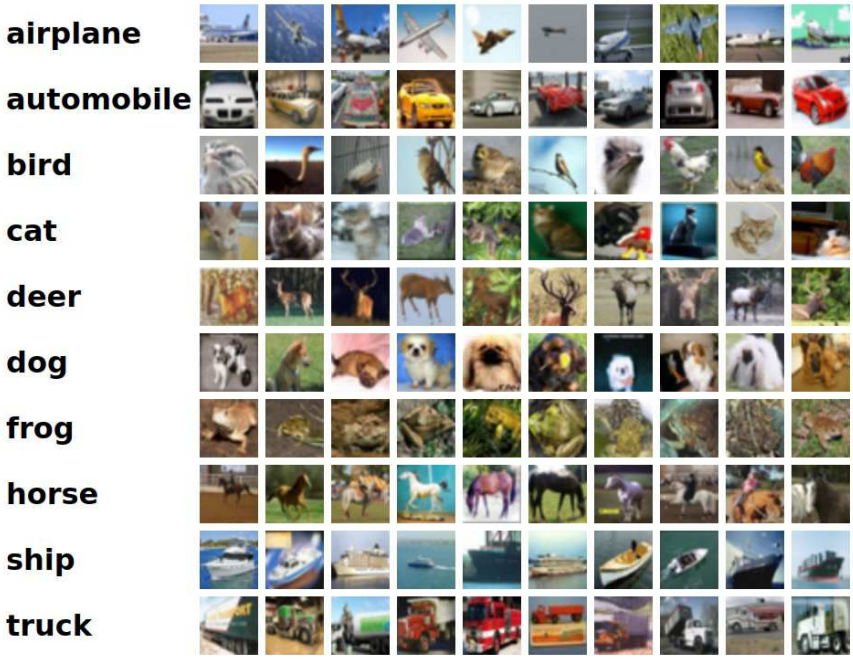
## 5.1. Pytorch

PyTorch je popularni okvir otvorenog koda za strojno učenje koji se široko koristi za istraživanje i razvoj u području dubokog učenja. Razvija ga i održava Linux Foundation i implementiran je u Pythonu. PyTorch pruža dinamički računalni graf koji omogućuje fleksibilnu i učinkovitu obradu gradijenata za propagaciju unatrag. Ova značajka olakšava implementaciju složenih modela te njihovo otklanjanje grešaka i modificiranje tijekom treniranja. PyTorch također pruža razne module i funkcije za konstrukciju neuronskih mreža, poput linearnih i konvolucijskih slojeva, aktivacijskih funkcija i funkcija gubitka. PyTorch podržava automatsku diferencijaciju, što znatno pojednostavljuje postupak izračunavanja gradijenata za propagaciju unatrag. PyTorch ima veliku i aktivnu zajednicu korisnika i razvijatelja te se kontinuirano razvija s novim značajkama i poboljšanjima.

## 5.2. Podatkovni skupovi

Skup podataka CIFAR-10 je često korištena referentna točka u području računalnog vida. Sastoji se od 60.000 32x32 slika u boji podijeljenih u 10 razreda, s 6.000 slika po razredu. Razredi obuhvaćaju avione, automobile, ptice, mačke, jelene, pse, žabe, konje, brodove i kamione. Skup podataka je podijeljen na skup za treniranje s 50.000 slika i skup za testiranje s 10.000 slika. Slike se predobrađuju oduzimanjem prosječne vrijednosti piksela za svaki kanal i dijeljenjem s standardnom devijacijom. Skup podataka je izazovan zbog malene veličine slika, niske rezolucije i visoke varijabilnosti u izgledu objekta, položaju i pozadini. Široko se koristi za evaluaciju

performansi modela klasifikacije slika i usporedbu različitih tehnika za proširivanje podataka, regularizaciju i optimizaciju.




Slika 4: Primjeri slika iz CIFAR-10 skupa podataka za različite klase unutar skupa podataka. Slika je preuzeta iz [23].

## 5.3. Detalji izvedbe

Naše istraživanje započelo je implementacijom originalnog Vision Transformer rada. Nakon toga, proširili smo naš rad kombinirajući ViT model s DenseNetom, stvarajući nekoliko različitih arhitektura kako bismo procijenili optimalan način organizacije i povezivanja transformer blokova u funkcionalne jedinice. Koristimo naziv transformer blok i sloj kao sinonime. Istražene su sljedeće varijante:

### 5.3.1. Vizualni transformer

Naša implementacija Vision Transformer rada poslužila je kao polazna točka za ovaj rad.

### 5.3.2. Vizualni transformer s težinama

Ulaz u određeni transformer blok je težinska suma svih prethodnih izlaza slojeva. Težinska suma je naučena i specifična za svaki sloj.

### 5.3.3. Vizualni transformer s težinama po značajkama

Ovaj pristup sličan je vizualnom transformeru s težinama, osim što se svi prethodni izlazi težinski ponderiraju po značajkama. Ako postoji n prethodnih slojeva s dimenzionalnošću d, ovaj pristup dodaje n*d naučenih parametara za svaki sloj.

### 5.3.4. Gusti vizualni transformer

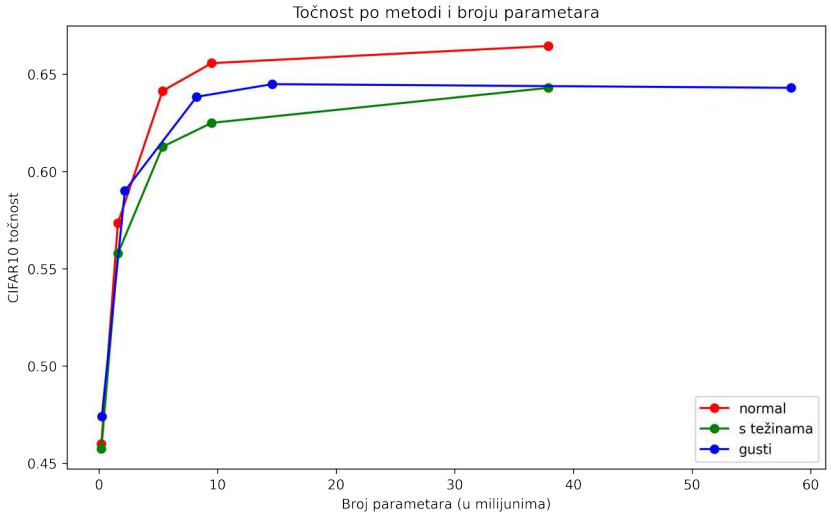
Inspiriran DenseNet gustim blokovima, svaki sloj u ovoj varijanti prima sve prethodne slojeve kao ulaz. Izlazi prethodnih slojeva se konkateniraju, a zatim se matrično množe kako bi se dobio potrebni ulazni dimenzionalitet za transformer blok. Ulazni i izlazni dimenzionaliteti transformer blokova su jednaki i konstantni kroz slojeve.

### 5.3.5. Gusti vizualni transformer nižeg ranga

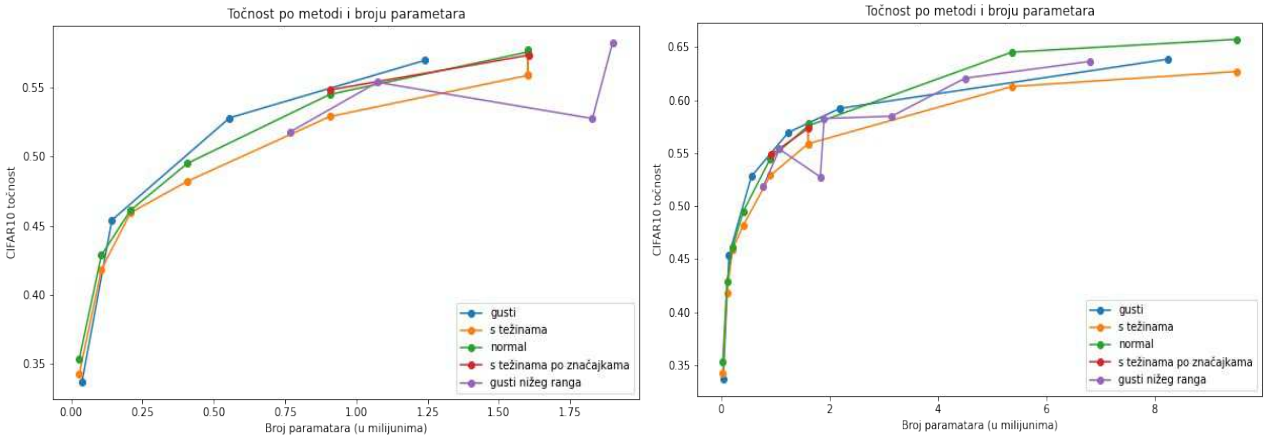
U ovoj varijanti pokušavamo smanjiti broj parametara u ukupnom modelu gustog transformera koristeći matricu nižeg ranga za linearnu transformaciju konkateniranih prethodnih izlaza. Prethodni izlazi se prvo projektiraju na nižu dimenzionalnost, a zatim se konkateniraju i transformiraju, kao u gustom transformeru.

# 6. Eksperimenti

Eksperimenti u ovom radu sastojali su se od tri faze, u prvoj fazi testirali smo manji broj varijanti na većem rasponu veličina modela, Slika 6., kako bismo odredili na kojim veličinama naše modifikacije pokazuju značajne razlike u odnosu na bazni model. U drugoj fazi smo za dobivene veličine testirali sve predložene modifikacije, a u trećoj mogućnoti treniranja modifikacije s najvećom točnošću.

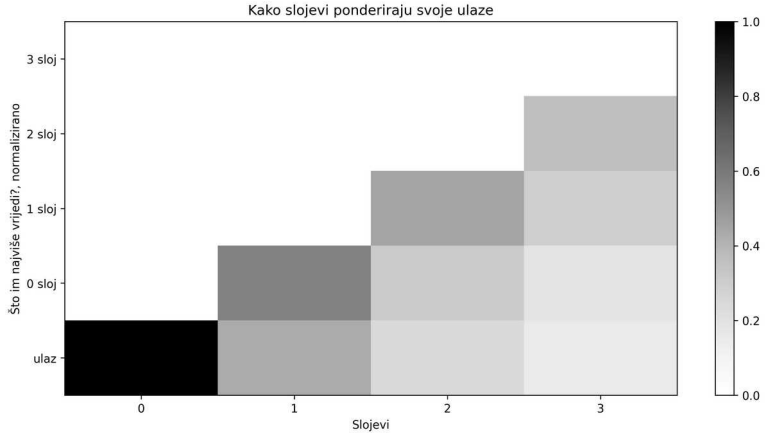



Slika 6: Prikaz točnosti na validacijskom setu za glavne klase predloženih arhitektura. Sa slike se vidi da je ne modificirani vizualni transformer pareto optimalniji za veće brojeve parametara, zato se u daljnoj analizi fokusiramo na vizualne transformere s brojem parametara koji odgovaraju redu veličine 1 milijun ili manje.

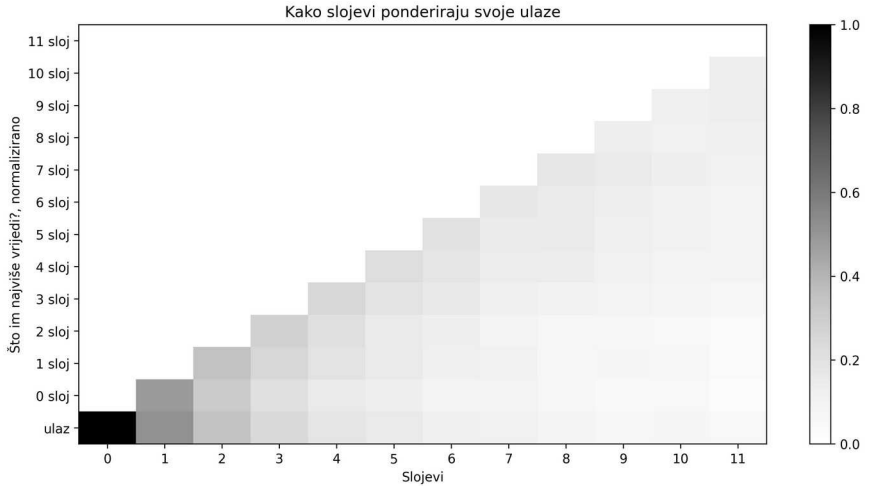



Slika 7: Prikaz točnosti različitih modificiranih arhitektura vizualnog transformera. Sa grafa je vidljivo da je gusti vizualni transformer pareto najoptimalniji za raspon parametara do 2 milijuna. Gusti vizualni transformer s težinama po značajkama ima točnost koja je blizu točnosti ne modificirane varijante. To objašnjavamo obzervacijom da svoje dodatne parametre postavlja tako da kopiraju izlaz prijašnjeg sloja.

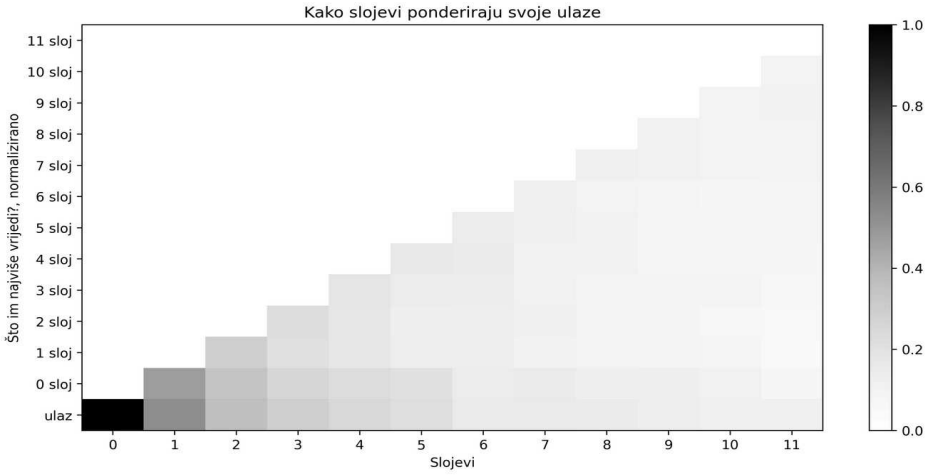
## 6.1. Vizualni transformer s težinama




Slika 8: Težine su normalizirane po stupcu. Sa slike se vidi da je najmanja varijanta vizualnog transformera s težinama naučila više težine stavljati na "dublje" slojeve.

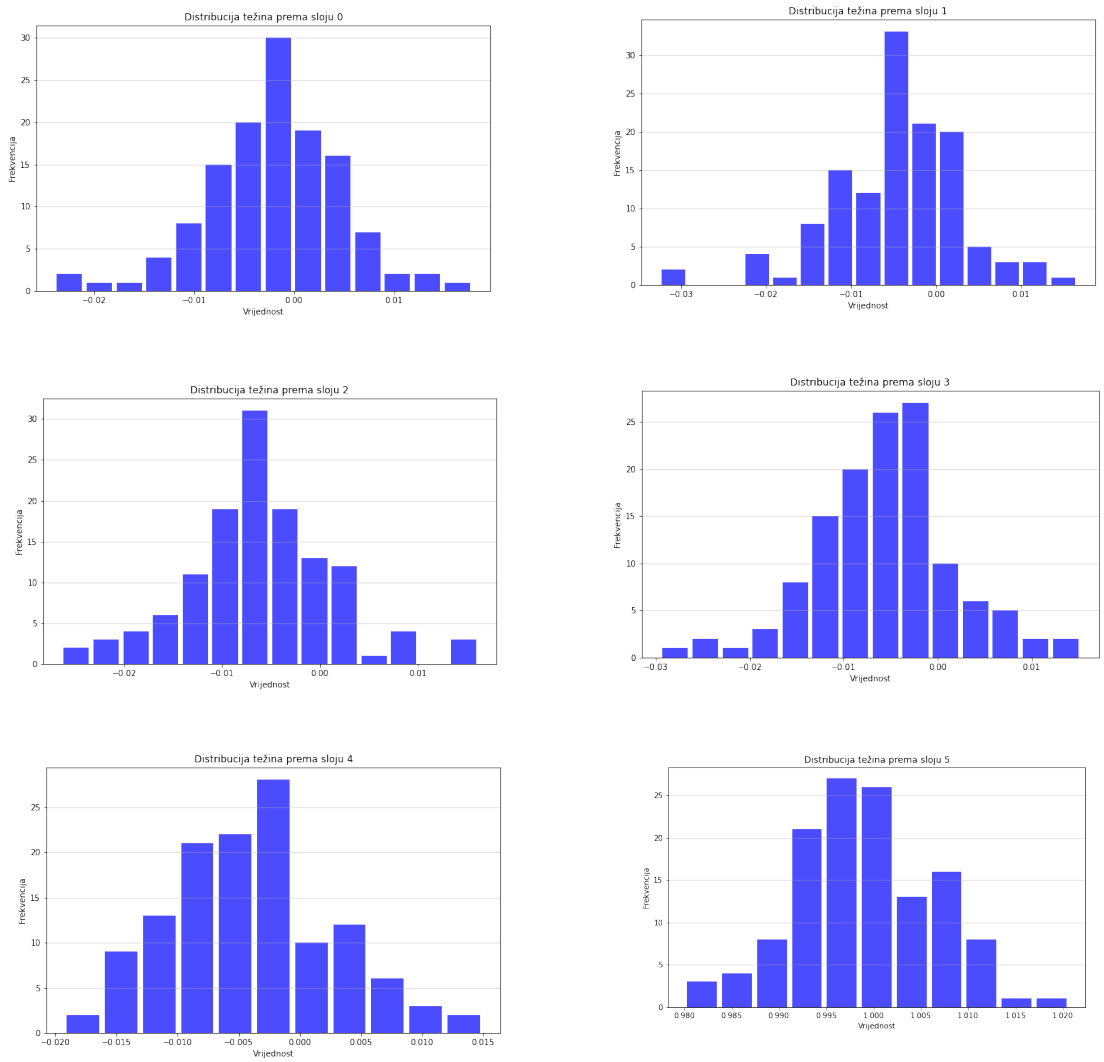



Slika 9: Distribucija težina za težinski vizualni transformer sa 9.5 milijuna parametara. Sa slike se vidi da model stavlja istu težinu na sve prijašnje slojeve.



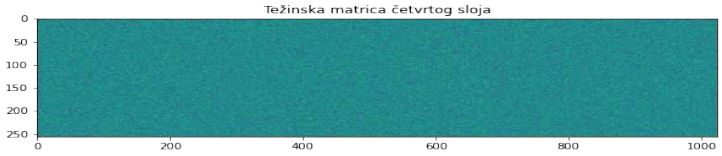

Slika 10: Distribucija težina za težinski vizualni transformer s 37.9 milijuna parametara. Kao i u 9.5 milijuna parametara varijanti model stavlja jednaku težinu na sve prijašnje slojeve. Težine su normalizirane po stupcu.

## 6.2. Vizualni transformer s težinama po značajkama




Slika 11: Distribucija težina prema pojedinim blokovima. Važno je napomenuti da je raspon vrijednosti za sve slojeve osim zadnjeg blizu 0. Model je naučio kopirati vrijednost prijašnjeg sloja. Analiza za 1.6 milijuna parametara model.

## 6.3. Gusti vizualni transformer




Slika 12: Prikaz matrice kombinacije izlaza prijašnjih slojeva u ulaz petog bloka gustog vizualnog transformera. Za razliku od težinskog i vizualnog transformera s težinama po značajkama ova matrica nema jednostavnu strukturu.

## 6.4. Svi eksperimenti

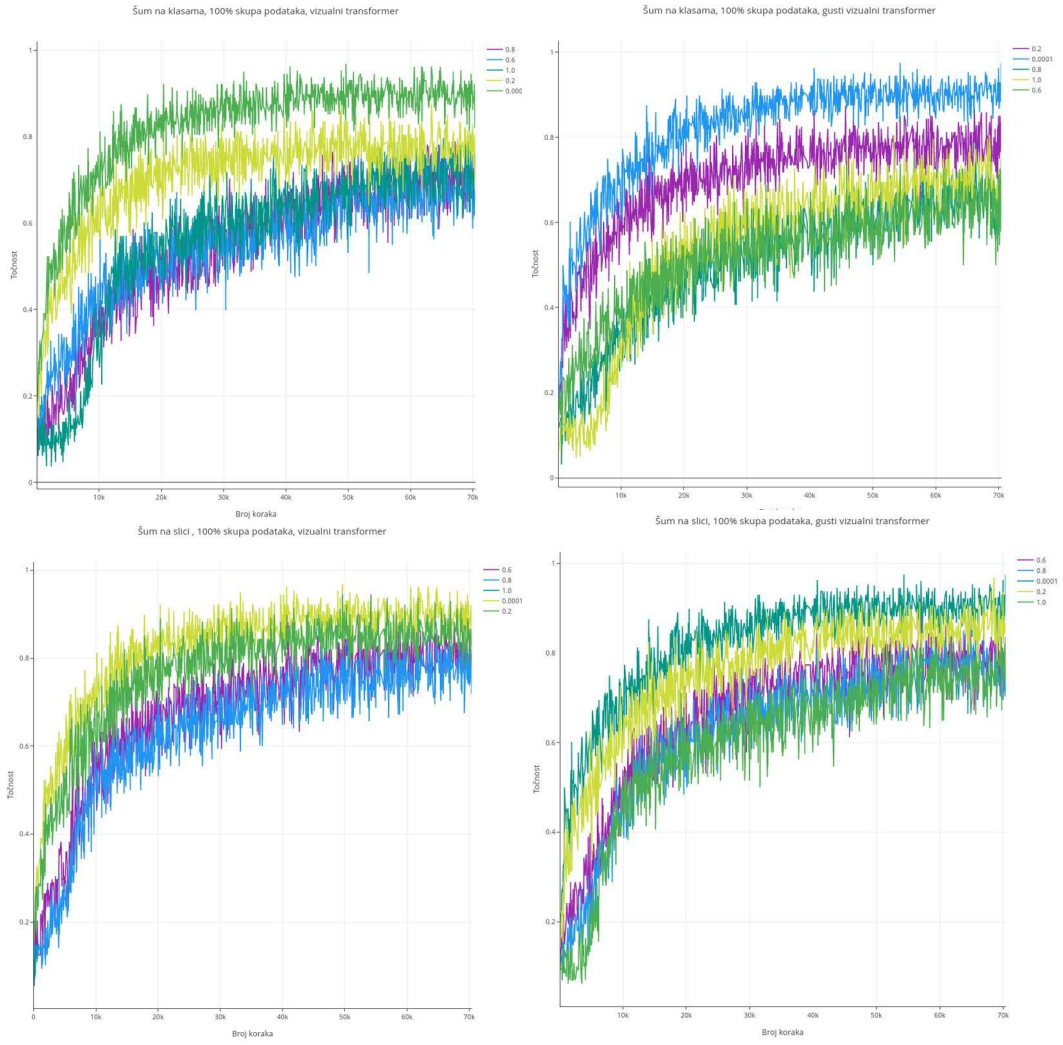
| Točnost | Veličina grupe | Komunikatori | d_model | Izlazavanje oznaka | Stopa učenja | Broj glava | Broj slojeva | Broj parametara/M | n_epoha | n_epoha_zaustavljavanja | Veličina pločica | Sjeme | Koristi raspoređivač | Raspad težine |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.6686 | 128.0 | obični | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 37.892 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6572 | 128.0 | obični | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 9.509 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6474 | 128.0 | gusti | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 14.624 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6452 | 128.0 | obični | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 5.362 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6448 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 37.892 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.643 | 128.0 | gusti | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 58.345 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6385 | 128.0 | gusti | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 8.24 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6365 | 128.0 | niskog ranga | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 6.80593 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6269 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 9.509 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6206 | 128.0 | niskog ranga | 156.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 4.5072092 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.6127 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 5.362 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5922 | 128.0 | gusti | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 2.192 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5922 | 128.0 | gusti | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 2.192906 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5848 | 128.0 | niskog ranga | 128.0 | 0.1 | 0.0001 | 8.0 | 12.0 | 3.145194 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5826 | 128.0 | niskog ranga | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.899146 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5775 | 128.0 | obični | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.602058 | 50.0 | 10.0 | 4.0 | 42.0 | True | 1e-06 |
| 0.5766 | 128.0 | težinski | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.602094 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5759 | 128.0 | obični | 128.0 | 0.0 | 0.0001 | 8.0 | 8.0 | 1.602058 | 50.0 | 10.0 | 4.0 | 42.0 | True | 1e-06 |
| 0.5759 | 128.0 | obični | 128.0 | 0.0 | 0.0001 | 8.0 | 8.0 | 1.602058 | 50.0 | 10.0 | 4.0 | 42.0 | True | 1e-06 |
| 0.5757 | 128.0 | obični | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.602058 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5757 | 128.0 | obični | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 1.602 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5757 | 128.0 | obični | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.602058 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5734 | 128.0 | po značajkama | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.606666 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5695 | 128.0 | gusti | 96.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.239178 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5587 | 128.0 | težinski | 128.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.602094 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5587 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 1.602 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5537 | 128.0 | niskog ranga | 96.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 1.074154 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5483 | 128.0 | po značajkama | 96.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.91009 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5449 | 128.0 | obični | 96.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.906634 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5289 | 128.0 | težinski | 96.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.90667 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.528 | 128.0 | gusti | 64.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.555786 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.5275 | 128.0 | niskog ranga | 64.0 | 0.1 | 0.0001 | 8.0 | 24.0 | 1.828298 | 50.0 | 4.0 | 4.0 | 42.0 | True | 1e-06 |
| 0.518 | 128.0 | niskog ranga | 64.0 | 0.1 | 0.0001 | 8.0 | 12.0 | 0.769514 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4951 | 128.0 | obični | 64.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.407818 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4821 | 128.0 | težinski | 64.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.407854 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4743 | 128.0 | gusti | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 58.345 | 200.0 | 10.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4611 | 128.0 | obični | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 0.208 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4594 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 0.208 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4594 | 128.0 | težinski | 192.0 | 0.1 | 0.0001 | 12.0 | 12.0 | 0.208 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.454 | 128.0 | gusti | 32.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.14273 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4285 | 128.0 | obični | 32.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.10561 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.4181 | 128.0 | težinski | 32.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.105646 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.3536 | 128.0 | obični | 16.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.028234 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.3428 | 128.0 | težinski | 16.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.02827 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |
| 0.3367 | 128.0 | gusti | 16.0 | 0.1 | 0.0001 | 8.0 | 8.0 | 0.037578 | 50.0 | 4.0 | 4.0 | 42.0 | True | 5e-05 |

Slika 13: Popis svih pokusa i njihovih hiperparametara poredanih u odnosu na točnost. Ne uključuje eksperimente sa šumom.

## 6.5. Eksperimenti sa šumom

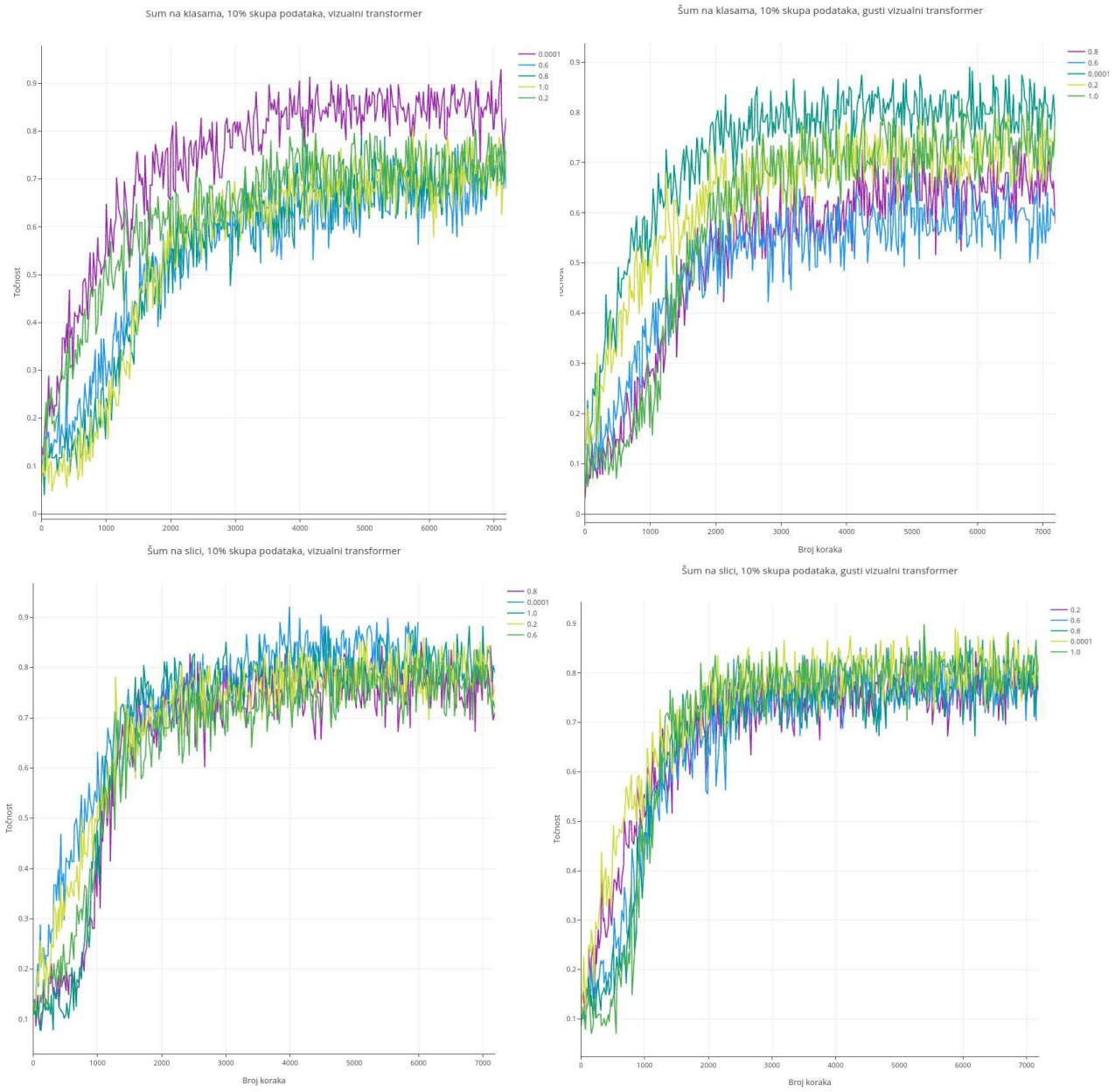
U sljedećim eksperimentima testiramo mogućnoti aproksimiranja vizualnog i gustog vizualnog transformera na različitim veličinama Cifar-10 skupa podataka. U svakom subgrafu variramo postotak skupa podataka koji je zamjenjen šumom. Broj epoha držimo konstantnim.

### 6.5.1. 100% skupa podataka



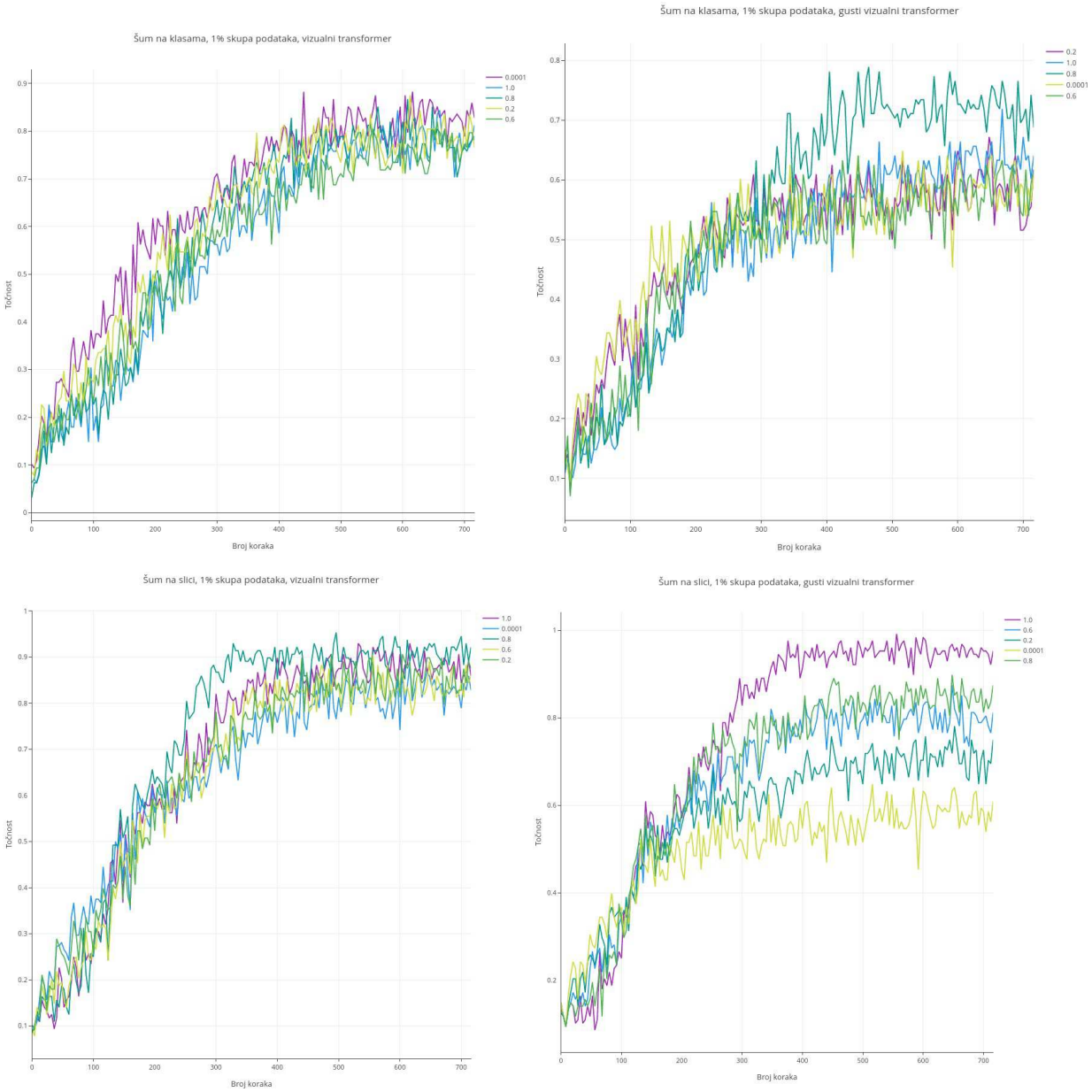

Slika 14: Prikaz evolucije treniranja pri cijelom Cifar-10 skupu podataka. Variramo razinu šuma. Pri treniranju na slikama koje se sastoje samo od šuma vizualni transformer uči brzo, nedostaje karakteristični sigmoidalni oblik. Gusti vizualni transformer uči bolje što je manje šuma.

### 6.5.2. 10% skupa podataka




Slika 15: Prikaz evolucije točnosti kroz treniranje modela. Koristi se 10% skupa podataka. Sa grafova se vidi da ukoliko se sve slike ili klase zamjene s onima sa šumom da model u početku sporije uči no doseže točnost modela koji imaju samo dio skupa podataka zamijenjen. Model s 60% skupa podataka zamjenjenog je najlošiji. Naša hipoteza je da 40% podataka iz kojega se može generalizirati dovodi do toga da radi više grešaka na dijelu skupa koji se sastoji od šuma.

### 6.5.3. 1% skupa podataka




Slika 16: Prikaz evolucije treniranja modela koji koristi 1% skupa podataka za različite razine šuma. Gusti vizualni transformer na ovoj veličini skupa podataka se lakše prenauči ukoliko mu je cijeli skup podataka zamjenjen šumom, u toj varijanti se nalaze i najveća odstupanja između razina šuma u konačnoj točnosti.

# 7. Rasprava

U ovom radu uveli smo četiri različite modifikacije vizualnog transformera, vizualni transformer s težinama, vizualni transformer s težinama po značajkama, gusti vizualni transformer te gusti vizualni transformer nižeg ranga.

Gusti vizualni transformer matrično množi konkatenirane prijašnje tokene te time dobiva ulaz u sljedeći sloj. On se pokazao pareto optimalnijim od vizualnog transformera bez modifikacija, s oko 1.6 milijuna parametara. Ostale modifikacije rezultiraju modelima koji su slabiji od osnovne opcije.

Gusti vizualni transformer se ponaša veoma slično vizualnom transformeru pri testiranju treniranja na različitim količinama šuma, osim u području malog broja primjera u skupu podataka. U tom rasponu lakše uči ukoliko je cijeli skup podataka sastavljen samo od šuma, što nas je iznenadilo te nismo uspijeli naći nikakav mehanizam koji bi objasnio takvo ponašanje. Pri 10% velićine skupa podataka, vizualni i gusti vizualni transformer najteže uče kada je broj podataka koji nisu i onih koji jesu zamjenjeni šumom jednak. Naša hipoteza je da model koristeći dio podataka koji nije zamjenjen ide generalizirati na podatke sa šumom te time ima manju točnost nad tim podacima.

S obzirom na veliki broj parametara matrice kombinacije prethodnih izlaza u gustom vizualnom transformeru, pokušali smo smanjiti veličinu te matrice, ali primijetili smo da gusti vizualni transformer s aproksimacijom manjeg ranga ne pruža bolji omjer broja parametara i konačne točnosti. Za razliku od drugih modifikacija, ova matrica i model nemaju upečatljiv oblik.

Vizualni transformer s težinama kao ulaz u sloj uzima naučenu težinu prijašnjih slojeva. U našim eksperimentima on pri manjim parametarskim brojevima stavlja najveću težinu na izlaze svih prethodnih slojeva. Time su jednako bitni, čime se replicira rezidualni sloj prisutan u transformerima. Ova varijanta ne doseže performanse ne modificiranog modela.

Model s težinama po značajkama uči težinsku sumu no po značajkama. U našim eksperimentima on je naučio postaviti dodatne parametre na način da zanemari sve izlaze slojeva osim izlaza neposredno prije sebe, što je vidljivo iz analize slike 11. Maksimalne vrijednosti težina dimenzija prvih slojeva su značajno manje od jedan, dok su sve vrijednosti težina prethodnog sloja blizu jedan. Razlog degradacije performansi vizualnog transformera s težinama po značajkama i težinskim vizualnim transformerom nismo pronašli, kao ni razlog za njihovu degradaciju u različite oblike.

Kako bi se dalje istražile ove pojave i razumjeli njihovi mehanizmi, potrebno je provesti dodatne analize i eksperimente. Mogući su daljnji istraživački koraci usmjereni prema razumijevanju utjecaja težinskih parametara na performanse vizualnih transformera te razvoju novih pristupa koji bi poboljšali omjer parametara i točnosti.

# 8. Zaključak

U sklopu ovog istraživanja bavili smo se analizom različitih metoda za povezivanje transformatorskih blokova vizualnog transformera. Naša studija ukazuje na to da gusto povezani vizualni transformatori predstavljaju pareto optimalnije rješenje u situacijama s ograničenim brojem parametara, u usporedbi s neizmijenjenim modelom. S druge strane, sve ostale modifikacije koje smo ispitivali rezultirale su gubitkom točnosti.

Modeli vizualnog transformatora s težinama i težinama po značajkama degradirali su u modele vrlo slične osnovnom neizmijenjenom modelu, unatoč većem kapacitetu, ovaj nalaz nas je iznenadio i trenutno nemamo objašnjenje za taj fenomen. Objašnjenje nemamo ni za ponašanje gustog vizualnog transformera za mali broj primjera. U tom području brže uči šum nego li podatke iz Cifar-10 skupa.

Kako bismo proširili naše istraživanje, potrebno je testirati hipotezu na modelima s većim brojem parametara, koristeći skupove podataka koji su bliži stvarnim primjenama. Ovo zahtijeva značajne računalne i vremenske resurse. Također, zbog ograničenja računalnih mogućnosti u ovom radu, nismo imali mogućnost prilagođavanja hiperparametara.

U budućim istraživanjima, preporučuje se provesti detaljniju analizu kako bismo bolje razumjeli fenomen i pružili dublje objašnjenje za rezultate koje smo dobili. Također, važno je istražiti i druge aspekte vizualnih transformatora te primijeniti složenije metode optimizacije kako bismo ostvarili napredak u performansama ovih modela.

# Literatura

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language Models are Few-Shot Learners.arXiv preprint arXiv:2005.14165.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, GeorgHeigold, Sylvain Gelly, Jakob Uszkoreit, i Neil Houlsby. An image is worth16x16 words: Transformers for image recognition at scale.arXiv preprint arXiv:2010.11929.

[3] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks.arXiv preprint arXiv:1608.06993

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, Judy Hoffman. Token Merging: Your ViT But Faster.arXiv preprint arxiv:2210.09461

[5] Hongxu Yin, Arash Vahdat, Jose Alvarez, Arun Mallya, Jan Kautz, Pavlo Molchanov. A-ViT: Adaptive Tokens for Efficient Vision Transformer.arxiv preprint arxiv:2112.07658

[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, . ImageNet classification with deep convolutional neural networks.Advances in neural information processingsystems, 25., stranice 1097-1105

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, i Li Fei-Fei.. Imagenet: Alarge-scale hierarchical image database..2009 IEEE Conference on ComputerVision and Pattern Recognition, stranice 248–255, 2009.

[8] Karen Simonyan, Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition.arXiv preprint arXiv:1409.1556

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Going Deeper with Convolutions.arXiv preprint arXiv:1409.4842

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition.arXiv preprint arXiv:1512.03385

[11] Mingxing Tan, Quoc V. Le.. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.arXiv preprint arXiv:1905.11946

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows.arXiv preprint arXiv:2103.14030

[13] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization.arXiv preprint arXiv:1412.6980

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning.arXiv preprint arXiv:1911.05722

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning.arXiv preprint arXiv:2006.07733

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, Ross Girshick. Masked Autoencoders Are Scalable Vision Learners.arXiv preprint arXiv:2111.06377

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.arXiv preprint arXiv:1810.04805

[18] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, Karen Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning.arXiv preprint arXiv:2204.14198

[19] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, Pete Florence. PaLM-E: An Embodied Multimodal Language Model.arXiv preprint arXiv:2303.03378

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language.arXiv preprint arXiv:1908.03557

[21] OpenAI. GPT-4 Technical Report.arXiv preprint arXiv:2303.08774

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Brad-bury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein,Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, ZacharyDeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, BenoitSteiner, Lu Fang, Junjie Bai, i Soumith Chintala. Pytorch: An impera-tive style, high-performance deep learning library..Advances in NeuralInformation Processing Systems 32, stranice 8024–8035. Curran Asso-ciates, Inc., 2019.

[23] Alex Krizhevsky, Vinod Nair, Geoffrey Hinton. Cifar-10 and cifar-100 datasets.URl: https://www. cs. toronto. edu/kriz/cifar. html, 6(1):1, 2009.

[24] Ilya Loshchilov, Frank Hutter. Decoupled Weight Decay Regularization.arXiv preprint arXiv:1711.05101

## Arhitekture za raspoznavanje slika slojevima pažnje


### Sažetak

Vizualni transformeri postali su standardna arhitektura za obradu vizualnih podataka. Obilježje transformerske arhitekture je rezidualni tok, koji olakšava optimizaciju modela, no restriktira izradu novih značajki jer ima fiksnu dimanzionalnost kroz cijeli model. Model treba odlučiti zadržati prethodno izračunate značajke ili dodati nove. U ovom radu se proučavaju četiri različite arhitektonske modifikacije vizualnog transformera s ciljem poboljšanja ponovne uporabe značajki i parametarske učinkovitosti. Inspiracija se crpi iz konvolucijskih modela, posebno DenseNet-a, koji je pokazao uspješno rješavanje sličnih problema. Provodimo eksperimente na CIFAR-10 skupu podataka, pri tom pokazujući da je jedna od predloženih varijanti, gusti vizualni transformer, pareto optimalnija od vizualnog transformera.



**Ključne riječi:** vizualni transformer, DenseNet, arhitektura, optimizacija


## Architectures for Image classification through attention layers


### Abstract

Vision transformers have become a standard architecture for processing visual data. The characteristic of transformer architecture is the residual flow, which facilitates model optimization but restricts the creation of new features due to fixed dimensionality throughout the model. The model needs to decide whether to keep the previously calculated features or add new ones. This paper investigates four different architectural modifications of the visual transformer with the aim of improving feature reuse and parameter efficiency. Inspiration is drawn from convolutional models, particularly DenseNet, which has successfully addressed similar problems. We conduct experiments on the CIFAR-10 dataset, demonstrating that one of the proposed variations, Dense Visual Transformer, is Pareto-optimal compared to the Vision Transformer.



**Keywords:** Vision Transformer, DenseNet, architecture, optimisation