

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

Lietišķo datorsistēmu institūts

Mākslīgā intelekta un sistēmu inženierijas katedra

Mārtiņš Jānis Ošmucnieks

bakalaura akadēmiskās studiju programmas "Robotizētas intelektuālas sistēmas"
students, stud. apl. nr. 211RDB322

**DEFORMĒJAMO KONVOLŪCIJU
MODIFIKĀCIJU SALĪDZINĀJUMS
OBJEKTU ATPAZĪŠANAS
UZDEVUMĀ**

BAKALAURA DARBS

Zinātniskais vadītājs PhD.sc.comp., pētnieks

Ēvalds Urtāns

RĪGA 2024

RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS
FAKULTĀTE

Lietišķo datorsistēmu institūts

Mākslīgā intelekta un sistēmu inženierijas katedra

Bakalaura darba izpildes lapa

Noslēguma darba autors:

students Mārtiņš Jānis Ošmucnieks

(paraksts, datums)

Noslēguma darbs ieteikts aizstāvēšanai:

Zinātniskais vadītājs:

PhD.sc.comp., pētnieks Ēvalds Urtāns

(paraksts, datums)

ANOTĀCIJA

Atslēgvārdi: Mašīnmācīšanās, dziļie neironu tīkli, konvolūcijas neironu tīkli, datorredze, deformējamās konvolūcijas, uzmanības mehānisms, objektu atpazīšana, attēlu klasifikācija

Bakalaura darba tips: Moderno risinājumu izpēte

Pēdējo 10 gadu laikā metodes, kas balstītas dziļajā mašīnmācīšanās, ir ieguvušas ļoti labus rezultātus dažādos datorredzes uzdevumos. Piemēram objektu atpazīšanas uzdevumā, kam ir neskaitāmi praktiski pielietojumi, konvolūcijas neironu tīkli uzrāda pārāku sniegumu. Šajā darbā tika aplūkotas deformējamās konvolūcijas, lai dziļāk izprastu to darbību un veiktu uzlabojumus. Galvenie pētījuma ieguldījumi ir sekojoši. Veikta sistemātiska literatūras analīze un tās kvantitatīvs un kvalitatīvs salīdzinājums, kurā tika apskatīti pētījumi, kas piedāvā jaunus dziļās mašīnmācīšanās mehānismus vai arhitektūras. Izveidota bieži izmantotās *ImageNet* datu kopas apakškopa *Small-ImageNet* ar ko var ātrāk veikts attēlu klasifikācijas eksperimentus. Veikti vairāki ablācijas pētījumi par deformējamo konvolūciju izmantošanu iekš *ResNet* arhitektūras. Arhitektūra, kas izmanto konvolūciju pirmajā slānī un deformējamās konvolūcijas pēdējos trīs iegūst visaugstākos rezultātus uzstādot 55.34% uz *Small-ImageNet* validācijas kopas. Izveidots un salīdzināts jauns mehānisms *LightDCN*, kas iedvesmojas no deformējamām konvolūcijām un samazina modeļa parametru par 6% skaitu saglabājot tā veiktspēju. Atvērtais pirmkods ir pieejams <https://github.com/marcho/deform-conv>.

ABSTRACT

Keywords: Machine learning, deep neural networks, convolution neural networks, computer vision, deformable convolutions, attention mechanism, object recognition, image classification

Bachelor thesis type: Research of modern solutions.

In the last 10 years, methods based in deep machine learning have achieved very good results in various computer vision tasks. For example, in the task of object detection, which has numerous practical applications, convolutional neural networks demonstrate superior performance. This work examined deformable convolutions to gain a deeper understanding of their operation and to make improvements. The main contributions of the research are as follows. A systematic literature review and its quantitative and qualitative comparisons were conducted, which examined studies proposing new deep machine learning mechanisms or architectures. A subset of the frequently used *ImageNet* dataset, *Small-ImageNet*, was created to facilitate faster image classification experiments. Several ablation studies on the use of deformable convolutions within the *ResNet* architecture were performed. The architecture that uses convolution in the first layer and deformable convolutions in the last three layers achieves the highest results, setting 55.34% on the *Small-ImageNet* validation set. A new mechanism, *LightDCN*, inspired by deformable convolutions was developed and compared, which reduces the number of model parameters by 6% whilst keeping the same performance. Source code is available at <https://github.com/march-o/deform-conv>.

SATURS

IEVADS	8
1. SAISTĪTIE PĒTĪJUMI	10
1.1. Dziļā mašīnmācīšanās	10
1.1.1. Mākslīgie neironu tīkli	10
1.1.2. Atpakaļpropogācijas algoritms	11
1.1.3. Nelinearitātes ieviešana ar aktivizēšanas funkcijām	13
1.2. Konvolūcijas neironu tīkli	14
1.2.1. Konvolūcijas mehānisms attēliem	15
1.2.2. Hierarhiskā konvolūcija	16
1.2.3. Iezīmju apvienošana	16
1.2.4. Dziļo datorredzes tīklu mehānismu evolūcija	17
1.3. Redzes transformatori	20
1.3.1. Uzmanības mehānisms	20
1.3.2. Redzes transformatoru arhitektūra	22
1.4. Deformējamās konvolūcijas	23
1.4.1. Deformējamo konvolūciju pamata ideja	24
1.4.2. Deformējamo konvolūciju uzlabojumi	25
2. SISTEMĀTISKĀ LITERATŪRAS ANALĪZE	27
2.1. Pētījumu meklēšanas protokols	27
2.2. Kvantitatīvs salīdzinājums	32
2.3. Kvalitatīvs salīdzinājums	33
3. METODOLOĢIJA	36
3.1. Datu kopas	36
3.2. Rādītāji	37
3.2.1. Klasifikācija	38
3.2.2. Semantiskā segmentācija	39
3.2.3. Objektu atpazīšana	40
3.3. Pētīto modeļu arhitektūras	42
3.3.1. Deformējamās konvolūcijas iekš <i>ResNet</i>	42
3.3.2. <i>InternImage</i> bloki iekš <i>Resnet</i>	42
3.3.3. Jaunieviestais mehānisms <i>LightDCN</i>	43
3.4. Apmācības un testēšanas protokols	44
3.4.1. Izmantotie optimizētāji un mācīšanās ātruma plānotāji	45
3.4.2. Mācīšanās ātruma atrašana	45

4. REZULTĀTI	47
4.1. Deformējamo konvolūciju modeļa apmācības atkārtošana	47
4.2. Eksperimenti attēlu klasifikācijas uzdevumā	48
4.2.1. <i>ResNet</i> ar deformējamajām konvolūcijām	48
4.2.2. <i>ResNet</i> ar <i>InternImage</i> blokiem	49
4.2.3. <i>LightDCN</i>	50
4.3. Eksperimenti objektu atpazīšanas uzdevumā	51
4.4. Kodolu deformējāciju apskats un salīdzinājums	51
SECINĀJUMI	53
IZMANTOTIE INFORMĀCIJAS AVOTI	54
PIELIKUMI	61

APZĪMĒJUMU SARAKSTS

ANN Mākslīgais neironu tīkls (angļu val. *Artificial neural network*). 10, 11, 12, 13, 14

BN Partiju normalizācija (angļu val. *Batch normalization*). 42

CNN Konvolūcijas neironu tīkls (angļu val. *Convolutional neural network*). 14, 16, 17, 18, 19, 49

Conv Konvolūcija (angļu val. *Convolution*). 24, 43

DCN Deformējamā konvolūcija (angļu val. *Deformable convolution*). 24, 42, 43, 44, 48, 49, 50, 51, 52, 53, 62

FFN Plūsmas pārsūtīšanas tīkls (angļu val. *Feed forward network*). 42

GPU Grafikas procesors (angļu val. *Graphics processing unit*). 18, 43, 44, 49

LN Slāņu normalizācija (angļu val. *Layer normalization*). 42

LR Mācīšanās ātrums (angļu val. *Learning rate*). 12, 45, 48, 62

MSE Vidējā kvadrātiskā kļūda (angļu val. *Mean squared error*). 12

NLP Dabiskās valodas apstrāde (angļu val. *Natural language processing*). 20

ViT Redzes transformators (angļu val. *Vision transformer*). 22, 42

IEVADS

Laikmetā, kad mums visapkārt ir dati, ir veikti ļoti daudz uzlabojumi to apstrādes metodoloģijā. No ekonomikas analīzes līdz ieteikumu sistēmām, pētnieki un izstrādātāji turpina attīstīt sistēmas, kas efektīvi izmanto lielus datu daudzumus, lai automatizētu un uzlabotu cilvēku darbu. Datoriem ir raksturīga īpašība labi apstrādāt skaitļus, citos domēnos to spēja atrast pamata sakarības un veidot precīzas hipotēzes ir ierobežota.

Specifiski vizuālos uzdevumos, kā attēlu klasifikācijā un detekcijā, automatizēta sistēma nestāv tuvu cilvēka spējām. Piemēram, bankas riska noteikšanas sistēma var aplūkot cilvēka pagātni, un nonākt pie precīzāka slēdziena nekā darbinieks. Tomēr sistēmas, kas nosaka automašīnas tehnisko stāvokli no vizuālās informācijas, netiek izmantotas, jo algoritmi, kas analizē attēlu, nav pietiekami precīzi, lai to rezultāti būtu noderīgi cilvēkiem.

Pēdējos gados datorredzes zinātne ir strauji attīstījusies, aktīvi katru gadu pārspējot nozīmīgus etalonuzdevumus un piedāvājot jaunas pieejas. Pirmais no darbiem, kas pavēra algoritmisku pieeju attēlu apstrādei, bija 1959. gada neirozinātnieku Dāvida Hūbela(*David Hubel*) un Torstena Vīzela(*Torsten Wiesel*) eksperimentālais pētījums (Hubel & Wiesel, 1959) par kaķu smadzeņu neironu atbildēm no vizuāliem stimuliem. Darbā tika atrasts, ka smadzenes apstrādā vizuālo informāciju neholistiskā manierē, ar to saprotot, ka pirms neironi identificē veselus objektus, vienkāršāki neironi atrod to komponentes kā stūrus un malas. Balstoties uz ideju par iezīmju atrašanu zinātnieks Kunihiro Fukušima(*Kunihiko Fukushima*) izveidoja teorētiski pirmo *dziļo* mākslīgo neironu tīklu (Fukushima, 1980). Viņa izveidotā sistēma izmantoja konvolūciju, slidinot svaru matricu pāri attēlam, tai *aktivizējoties*, kad svarīgās iezīmes ar to pārklājas, tādā veidā iezīmējot nozīmīgus reģionus. 1989. gadā franču zinātnieks Jans LeKuns (*Yann LeCun*) uzlaboja Fukušimas sistēmu, pievienojot tai atpakaļizplatīšanās(angļu val. *backpropagation*) (Rumelhart, Hinton et al., 1986) algoritmu. Tā tika izveidots pirmais konvolūcijas neironu tīkls(Y. LeCun, Boser et al., 1989).

Palielinoties datorredzes zinātnes komūnai tika izveidotas vairākas standartizētas datu kopas ar kurām tika salīdzināti piedāvātie algoritmi. Viens no populārākajiem etalonuzdevumiem *ImageNet*(Deng, Dong et al., 2009), līdzīgi kā citi, ietur ikgadējas sacensības salīdzinot algoritmu attēlu klasifikācijas precizitāti. Ja iepriekšējās metodes, kas balstījās iezīmju meklēšanā ar klasisku algoritmisku pieeju, spēja precīzi klasificēt attēlus ar 26 procentu kļūdu, 2012. gadā viss mainījās, kad konvolūcijas neironu tīkls *AlexNet*(Krizhevsky, Sutskever et al., 2012) ieguva 16 procentu kļūdu. Šis bija ļoti nozīmīgs uzlabojums konvolūcijas tīkliem

un mašīnmācīšanās jomai kopumā.

Kopš *AlexNet* izveidošanas *ImageNet* kļūda ir nokritusies zem desmit procentiem. Visi nākamie uzvarētāji bijuši arvien jauni konvolūcijas neironu tīkli. Viens no konvolūcijas veidiem - deformējamās konvolūcijas, pēdējos gados ir redzējis labus rezultātus vairākos datorredzes uzdevumos kā objektu atpazīšanā. Savā darbā dziļāk aplūkošu tieši šo attēlu apstrādes mehānismu.

Darba mērķis ir atkārtot deformējamo konvolūciju rezultātus objektu atpazīšanas un klasifikācijas uzdevumā un testēt arhitektūras un apmācības metodoloģijas izmaiņu ietekmi uz risinājuma veikspēju. Ir izvirzīti sekojošie darba uzdevumi:

1. Apgūt dziļajā mašīnmācīšanā balstītās objektu atpazīšanas un klasifikācijas sistēmas.
2. Veikt sistemātisku literatūras analīzi.
3. Atrast datu kopas, kuras būtu piemērotas objektu atpazīšanai un klasifikācijai.
4. Atrast un apgūt rādītājus ar kurām noteikt modeļu veikspēju.
5. Apgūt un atkārtot rezultātus, izmantojot deformējamo konvolūciju modeli.
6. Veikt izmaiņas attēlu klasificēšanas un objektu atpazīšanas modeļiem, pievienojot deformējamās konvolūcijas, un tās novērtēt.
7. Eksperimentāli salīdzināt deformējamo konvolūciju modeļus.

1. SAISTĪTIE PĒTĪJUMI

Šajā nodaļā tiks aprakstīti dziļās mašīnmācības un konvolūcijas neironu tīklu pamatprincipi. Turpinājumā tiks aplūkoti svarīgākie arhitektūras uzlabojumi konvolūciju tīkliem, kā arī moderni attēlu apstrādes modeļi, kas neizmanto konvolūcijas, bet uzmanības mehānismu. Nobeigumā tiks veikts padziļināts ieskats deformējamajās konvolūcijās un to uzlabojumos.

1.1. Dziļā mašīnmācīšanās

Dziļā mašīnmācīšanās ir metožu kopa, kas izmanto mašīnmācīšanos un mākslīgos neironu tīklus, lai izpildītu noteiktus kvantificējamus uzdevumus. Tā veitā, lai izmantotu algoritmisku pieeju uzdevumu pildīšanai, kur pētnieks vai eksperts izveido sistēmu balstoties uz savām zināšanām par attiecīgo jomu, dziļā mašīnmācīšanās paļaujas uz lieliem datu daudzumiem un dziļiem neironu tīkliem. Eksistē trīs galvenās mašīnmācīšanās metodes. Šajā darbā tiks aplūkota viena no tām - uzraudzīta mašīnmācīšanās.

- Uzraudzīta mācīšanās izmanto ieejas un izejas datus, lai modelētu to sakarības, minimizējot atšķirību starp īstajiem izejas signāliem un sintezētajiem.
- Pašuzraudzīta jeb vājā uzraudzītā mācīšanās izmanto mazu ieejas un tiem atbilstošo izejas datu daudzumu, viegli modelējot to sakarības. Papildus tiek izmantots liels daudzums ieejas datu, kuriem nav atbilstošu izejas datu. Tādā veidā tiek iegūtas dziļākas sakarības starp ieejas datiem. Rezultātā sistēma tiek uzlabota.
- Neuzraudzīta mācīšanās izmanto tikai ieejas datus, samazinot to dimensionalitāti jeb kompresējot un atdarinot līdzīgus paraugus.

1.1.1 Mākslīgie neironu tīkli

Mākslīgie neironu tīkli(ANN) ir matemātiski modeļi, kas atveido bioloģisku neironu tīklu darbību. To galvenā sastāvdaļa ir svāri jeb parametri, kas līdzīgi sinapsēm smadzenēs ietekmē izejošo impulsu aktivitāti balstoties uz ieejas signālu. Neironu tīklus var raksturot ar četrām īpašībām - ieejas un izejas signālu skaits, slāņu skaits un slāņu neironu skaits. Viena slāņa ANN ar diviem ieejas un vienu izejas signālu var aprakstīt ar sekojošo formulu (Goodfellow, Bengio et al.,

2016):

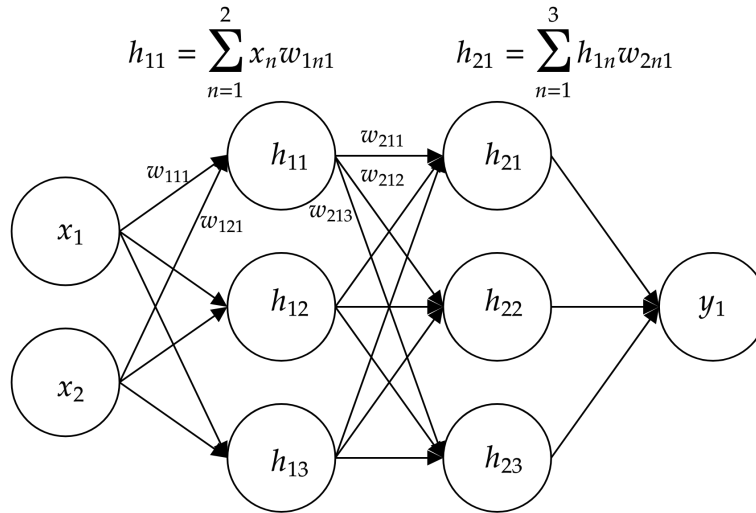
$$\begin{aligned} h_j &= \sum_{n=1}^2 X_n W_{i_{jn}} \\ y &= \sum_{j=1}^n h_j W_{o_j} \end{aligned} \quad (1.1)$$

Kur $X \in \mathbb{R}^2$ ir ieejas signāla vektors, h_i ir katra neirona vērtība un n neironu skaits. $W_i \in \mathbb{R}^{2 \times n}$ ir svāri starp ieejas vektoru un neironiem, $W_o \in \mathbb{R}^{n \times 1}$ starp neironiem un izeju. Izejas vektoru $Y \in \mathbb{R}^1$ iespējams aprēķināt arī šādi:

$$Y = X \times W_i \times W_o \quad (1.2)$$

Diagrammā 1.1. ir attēlots mākslīgais neironu tīkls ar diviem ieejas, vienu izejas signālu un diviem slāņiem, kuri ir trīs neironus dziļi.

Pirmo reizi ANN īstenoja Frenks Rosenblats (*Frank Rosenblatt*) 1958.gadā (Rosenblatt, 1958), izveidojot *perceptronu*. *Perceptronam* bija 400 ieejas signālu, ko veidoja 20 reiz 20 fotoelementu režģis, viens slānis ar 512 neironiem un astoņi izejas signāli. Tā mērķis bija attēlu atpazīšana. Mūsdienās lielākie mākslīgie neironu tīkli sastāv no vairākiem triljoniem svaru, un tiem laika gaitā pievienoti neskaitāmi dažāda veida uzlabojumi.



1.1. att. Mākslīgā neironu tīkla shēma

1.1.2 Atpakaļpropagācijas algoritms

Lai mākslīgs neironu tīkls spētu precīzi modelēt sakarības starp ieejas un izejas signāliem, tā svāri jeb parametri tiek pielāgoti izmantojot gradienta

lejupslīdes metodi. Atpakaļpropogācijas algoritms ir gradienta lejupslīdes metodes efektīvs pielietojums. Tā saknes ir atrodamas 1673.gadā Leibnīca ķēdes likumā, pirmo reizi tas tika īstenots mākslīgajos neironu tīklos 1986.gadā (Rumelhart, Hinton et al., 1986). Tika parādīts, ka izmantojot šādu algoritmu strādājot ar īstiem datiem, ANN savos neironos tur noderīgus iezīmju vektorus. Šī metode optimizē neironu tīkla svarus, minimizējot kļūdu starp tīkla prognozēto un patieso izejas vērtību.

Galvenā algoritma ideja un tā izvērsta aplūkošana tiek ņemta no (Rumelhart, Hinton et al., 1986) un (Goodfellow, Bengio et al., 2016). Kļūdas aprēķināšanai parasti tiek izmantota zuduma funkcija, piemēram, Vidējā kvadrātiskā kļūda (angļu val. *Mean squared error*), kas definēta kā:

$$\mathcal{L}_{MSE} = (y - y')^2 \quad (1.3)$$

Kur y' ir patiesā izejas vērtība, bet y - prognozētā vērtība. Kā piemēru varam izmantot iepriekš minēto ANN ar vienu slāni. Aplūkojot formulas (1.1.) un (1.3.), varam atrast ka atvasinājums no kļūdas attiecībā pret prognozēto izejas vērtību ir:

$$\frac{\partial \mathcal{L}_{MSE}}{\partial y} = 2(y - y') \quad (1.4)$$

Un atvasinājums no prognozētās vērtības attiecībā pret svaru W_{oj} :

$$\frac{\partial y}{\partial W_{oj}} = h_j \quad (1.5)$$

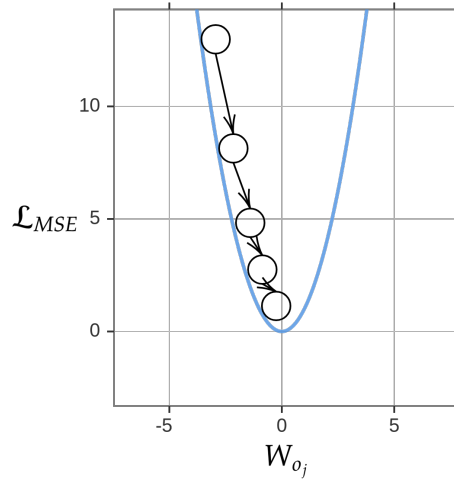
Pielietojot ķēdes likumu uz formulām (1.4.) un (1.5.), varam iegūt atvasinājuma no kļūdas attiecībā uz svāriem:

$$\frac{\partial \mathcal{L}_{MSE}}{\partial W_{oj}} = \frac{\partial \mathcal{L}_{MSE}}{\partial y} \cdot \frac{\partial y}{\partial W_{oj}} = 2(y - y') \cdot h_j \quad (1.6)$$

Līdzīgi iespējams atrast atvasinājumu kļūdai attiecībā pret ieejas svaru W_{ijn} , izmantojot ķēdes likumu un formulu (1.5.):

$$\frac{\partial \mathcal{L}_{MSE}}{\partial W_{ijn}} = \frac{\partial \mathcal{L}_{MSE}}{\partial y} \cdot \frac{\partial y}{\partial h_j} \cdot \frac{\partial h_j}{\partial W_{ijn}} = 2(y - y') \cdot W_{oj} \cdot X_n \quad (1.7)$$

Lai efektīvi pielāgotu ANN svarus ir jāizveido algoritms, kur iteratīvā veidā tiek aprēķināta kļūda, svaru gradienti pret kļūdu, un svaru vērtības tiek modificētas tā, lai kļūda samazinātos, tas ir pretējā virzienā no atvasinājuma. Tiek izmantots hiperparametrs η (LR), kas nosaka cik ļoti svāri tiks modificēti katrā iterācijā, lai *nepāršautu* pāri optimālajai vērtībai. Naivas implementācijas



1.2. att. Gradianta lejupslīdes ilustrācija

pseudokods redzams algoritmā 1.1., un intuitīva ilustrācija par to kā tiek modificēti ANN svāri balstoties uz kļūdu un tās atvasinājumu redzama attēlā 1.2..

1.1. algoritms Svaru atjaunināšana izmantojot gradianta lejupslīdi neironu tīklā (balstīts uz (Goodfellow, Bengio et al., 2016))

- 1: **while** nav sasniegts konverģences kritērijs **do**
- 2: **for** katram mācību piemēram **do**
- 3: Aprēķināt tīkla izvadi izmantojot pašreizējos svarus (1.2.)
- 4: Aprēķināt kļūdas funkciju izmantojot (1.3.)
- 5: Aprēķināt gradientu attiecībā pret katra slāņa svāriem izmantojot (1.6.) un (1.7.)
- 6: Modificēt izvades svarus:

$$W_{o_j} := W_{o_j} - \eta \cdot \frac{\partial \mathcal{L}_{MSE}}{\partial W_{o_j}}$$

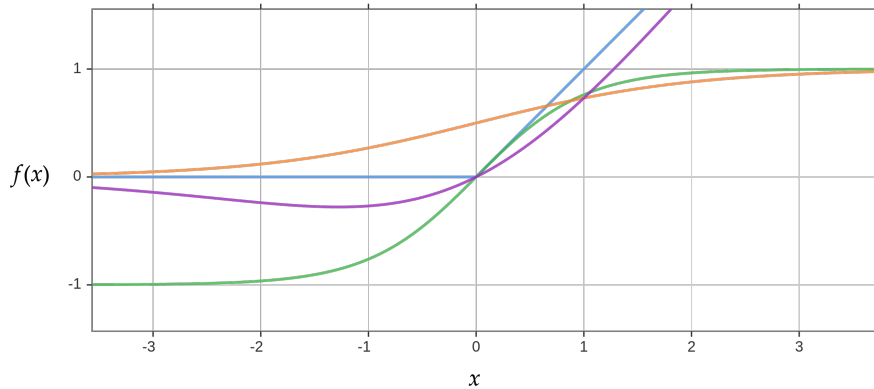
- 7: Modificēt ieejas svarus:

$$W_{i_{jn}} := W_{i_{jn}} - \eta \cdot \frac{\partial \mathcal{L}_{MSE}}{\partial W_{i_{jn}}}$$

1.1.3 Nelinearitātes ieviešana ar aktivizēšanas funkcijām

Pagaidām izveidotais ANN var modelēt tikai lineāras attiecības starp ieejas un izejas datiem, jo, lai arī cik daudz slāņu tam tiek pievienoti, to var reducēt uz lineāru vienādojumu. Tādēļ starp neironiem nepieciešams pievienot nelineāras funkcijas - aktivizācijas funkcijas, kas var modelēt kompleksākas attiecības starp ieejas un izejas datiem. Attēlā 1.3. redzamas dažādas bieži izmantotas aktivizācijas funkcijas:

- oranžā - $Sigmoid(x) = \frac{1}{1+e^{-x}}$
- zaļā - $Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- zilā - $ReLU(x) = \max(0, x)$
- violetā - $SiLU(x) = \frac{x}{1+e^{-x}}$



1.3. att. Dažādas aktivizēšanas funkcijas (ņemtas no (Goodfellow, Bengio et al., 2016))

Pirmo reizi aktivizēšanas funkcijas tika izmantotas Frenka Rosenblatta *perceptronā* (Rosenblatt, 1958). Iedvesmojoties no bioloģiskiem neironiem, šis modelis izmantoja parastu soļa funkciju, lai noteiktu vai attiecīgais mākslīgais neirons sūta savu signālu tālāk:

$$f(x) = \begin{cases} 0, & \text{ja } x < 0 \\ 1, & \text{ja } x \geq 0 \end{cases} \quad (1.8)$$

1969.gadā Kīnīhiko Fukušima (Fukushima, 1969) savā konvolūcijas neironu tīklā ieviesa *ReLU* (taisngrieža lineārās vienības) aktivizēšanas funkciju (attēls 1.3.). Mūsdienās *ReLU* un tās varianti, kā *GeLU* (Hendrycks & Gimpel, 2016), ir visbiežāk izmantotās aktivizācijas funkcijas dziļajos neironu tīklos priekš attēlu apstrādes.

1.2. Konvolūcijas neironu tīkli

Konvolūcijas neironu tīkls (CNN) ir ANN veids, ko izmanto dažādos ar attēliem saistītos un citos uzdevumos. Tam ir daudz mazāk parametru un ir skaitliski mazāk sarežģīts. To galvenā sastāvdaļa ir konvolūcijas mehānisms, kas aizvieto pilnībā savienotus neironu slāņus, un izmanto filtru ideju, lai attēlā atrastu svarīgākās īpašības.

1.2.1 Konvolūcijas mehānisms attēliem

Konvolūcijas mehānisms signālu procesēšanā tiek izmantots jau kopš 18. gadsimta. Viens no pirmajiem konvolūcijas integrāļa izmantošanas veidiem parādījās D' Alemberta Teilora teorēmas atvasinājumā, izteikts izteiksmē:

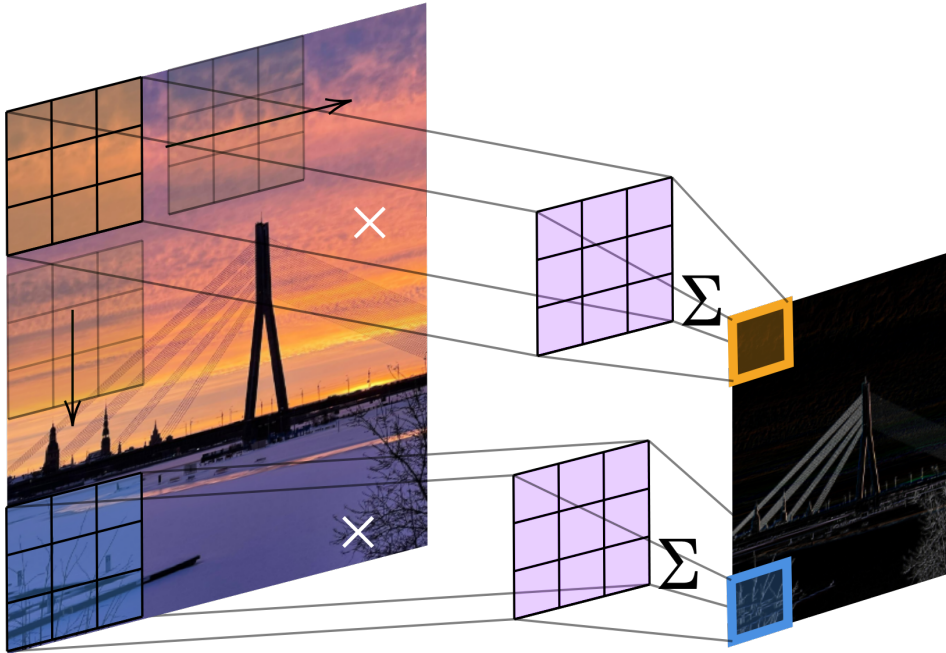
$$\int f(u) \cdot g(x - u) du \quad (1.9)$$

Būtībā konvolūcija ir divu funkciju reizinājuma integrālis laikā, kad viena funkcija tiek *slidināta* pāri otrai. Šis princips tika daudz izmantots Furjē un Laplasa darbos, ar ideju par to, ka divu signālu spektru reizinājums ir vienāds ar signālu konvolūcijas spektru.

Attēlu apstrādē tiek izmantota 2 dimensiju diskrētā konvolūcija (Goodfellow, Bengio et al., 2016):

$$(f * g)(x, y) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) \cdot g(x - i, y - j) \quad (1.10)$$

Šī operācija jau sen tiek izmantota dažādos attēlu apstrādes uzdevumos, kā izplūdināšana un asināšana. Attēlā 1.4. redzama konvolūcijas shēma. Tiek veikta divdimensiju konvolūcija starp attēla matricu un mazāku matricu ar koeficientiem - *kerneli* priekš malu atrašanas.

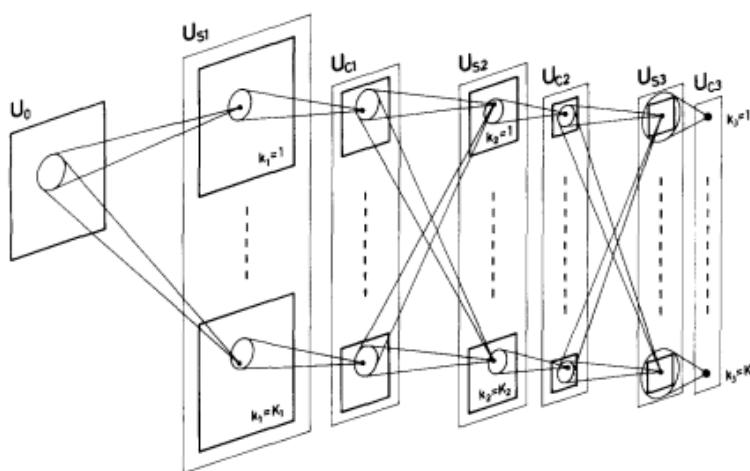


1.4. att. Attēlu konvolūcijas shēma

1.2.2 Hierarhiskā konvolūcija

Pēc (Hubel & Wiesel, 1959) atradumiem, Fukušima(Fukushima, 1980) veidoja savu CNN balstoties uz domu par augošu kompleksitātes līmeņu šūnu savienojumu, redzamu 1.5. attēlā.

Šāda tipa arhitektūra ir stūrakmens moderniem konvolūciju neironu tīkliem. Pirmo, seklāko kerneļu svāri ir tā izvietoti, lai meklētu prastākās iezīmes - malas, stūrus. Tiek izveidota karstuma karte par vietām, kur atrodas vēlamās iezīmes. Šī karstuma karte ir mazākā telpiskā izšķirtspējā, un katrs pikselis ietver informāciju par veselu pikseļu apkaimi. Tiek atkārtots konvolūcijas process, un iegūtas augstākas abstrakcijas iezīmes - jau veselas objektu daļu atrašanās vietas.



1.5. att. Neocognitron uzbūves salīdzinājums ar redzes smadzeņu šūnām (aizgūts no (Fukushima, 1980))

1.2.3 Iezīmju apvienošana

Iezīmju apvienošana jeb *pooling* vai apakšparaugošana(*subsampling*) ir būtisks sastāvdaļa no konvolūcijas neironu tīkliem. Izmantota priekš attēlu samazināšanas, konvolūcijas neironu tīklos to ievietoja (Y. LeCun, Boser et al., 1989), izmantojot šo mehānismu pēc konvolūcijas. *Pooling* darbība ir vienkārša - kāds attēla reģions tiek kompresēts uz mazāka izmēra reģionu, paturot tikai svarīgāko informāciju. Pirmkārt iezīmju apvienošana tiek izmantota, lai samazinātu modeļa skaitļošanas kompleksitāti. Otrkārt šī metode idejiski no kāda reģiona apkopo svarīgākās iezīmes, šis seko idejai par informācijas izgūšanu ar dimensiju samazināšanu, par ko ir aprakstīt iepriekšējā nodaļā. Ir divi visbiežāk izmantotie iezīmju apvienošanas varianti:

AveragePooling Vidējā iezīmju apvienošana reģionu kompresē uz tā skaitliski vi-

dējo vērtību. Izmantots senākos pētījumos, kā arī attēlu samazināšanā.

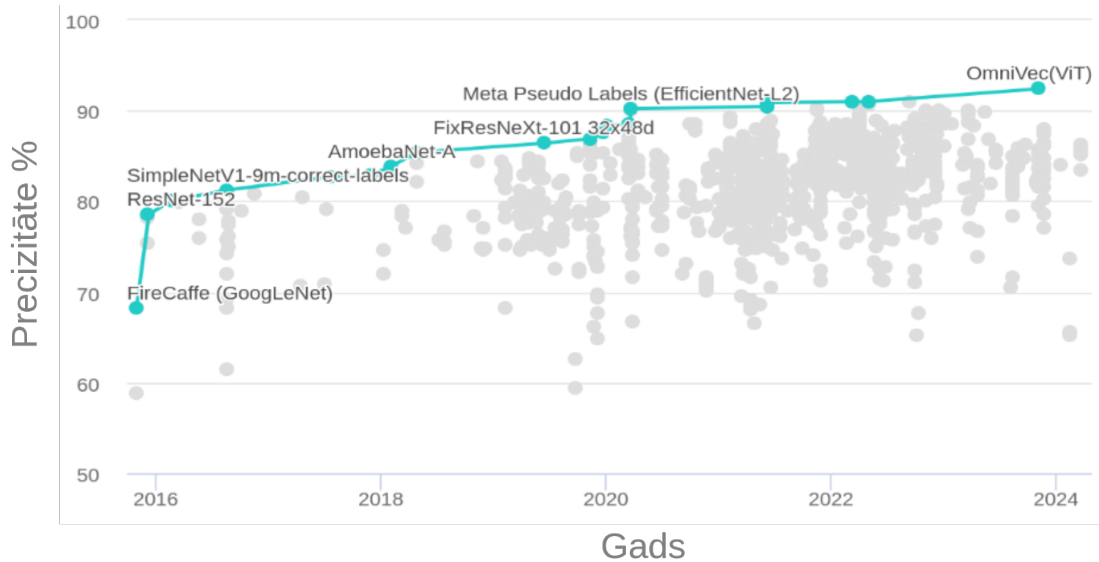
$$\begin{bmatrix} 1 & 3 & 2 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 8 & 7 & 6 \\ 5 & 3 & 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 3.75 & 5.25 \\ 6.25 & 4 \end{bmatrix}$$

MaxPooling Maksimuma iezīmju apvienošana reģionu kompresē uz skaitliski lielāko vērtību. Modernāks paņēmieni, kas ir piemērotāks konvolūcijas neironu tīkliem.

$$\begin{bmatrix} 1 & 3 & 2 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 8 & 7 & 6 \\ 5 & 3 & 2 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 6 & 8 \\ 9 & 7 \end{bmatrix}$$

1.2.4 Dziļo datorredzes tīklu mehānismu evolūcija

Kopš *AlexNet* (Krizhevsky, Sutskever et al., 2012) izveides 2012. gadā, CNN ar katru gadu mainās un uzlabojas, attēls 1.6.. Šajā nodaļā aplūkošu trīs arhitektūras, kas visvairāk pacēla *ImageNet* (Deng, Dong et al., 2009) uzdevuma precizitātes līmeni un izveidoja pamata metodes kā veidot augstas veiktspējas dziļos konvolūcijas neironu tīklus priekš attēlu apstrādes.

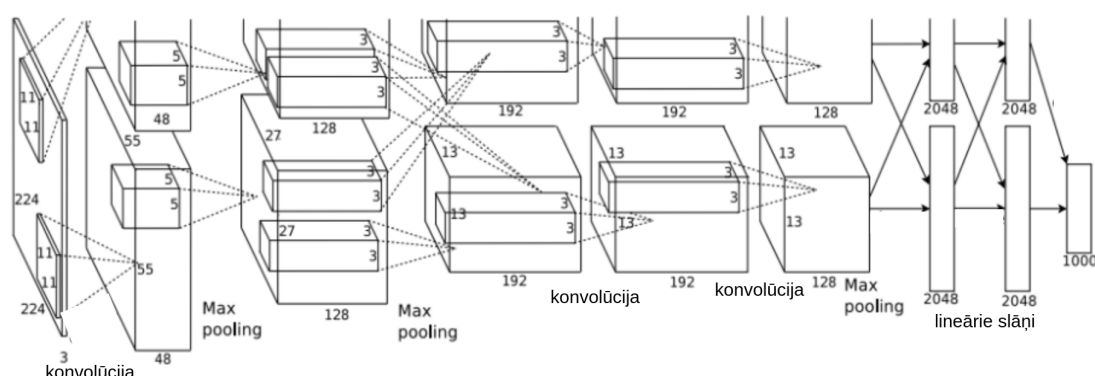


1.6. att. *ImageNet* etalonuzdevuma precizitāte pa gadiem (modificēts no (P. contributors, 2017))

AlexNet Aleksa Križevska (*Alex Krizhevsky*), Iljas Sutskevera (*Ilya Sutskever*) un Žofrē Hintonā (*Geoffrey E. Hinton*) izstrādātais CNN demonstrēja būtisku

uzlabojumu (10.8% zemāku top-5 kļūdu) pār iepriekšējajām tradicionālajām metodēm un atjaunoja pasaules interesi dziļajos konvolūcijas tīklos. Nozīmīgākie uzlabojumi bija:

- *ReLU* aktivizēšanas funkciju izmantošana, kas būtiski samazina tīkla trenēšanas laiku saglabājot precizitāti, ļaujot pētniekiem veikt ievērojami vairāk eksperimentus.
- Trenēšana uz diviem GPU. Tīkls tika sadalīts divās daļās un katras puses izeja un atvasinājumi tika rēķināti uz attiecīgās ierīces. Šis iedeva minimālu pāātrinājumu.
- Pārklājošos iezīmju apvienošana, kas samazināja top 1 kļūdu par 0.4% un samazināja pārmērīgu pielāgošanos treniņkopai.
- Lai samazinātu pārmērīgu pielāgošanos treniņkopai(angļu val. *overfitting*), treniņu kopa tiek palielināta ar attēlu translācijas un krāsu augmentācijām. Kā arī tiek pievienota neironu atskaite(angļu val. *dropout*), kas stohastiski daļu no neironiem nonullē, tādējādi veidojot robustākus svarus, kas saglabā augstu precizitāti uz neredzētiem piemēriem.



1.7. att. *AlexNet* arhitektūras shēma (modificēts no (Krizhevsky, Sutskever et al., 2012))

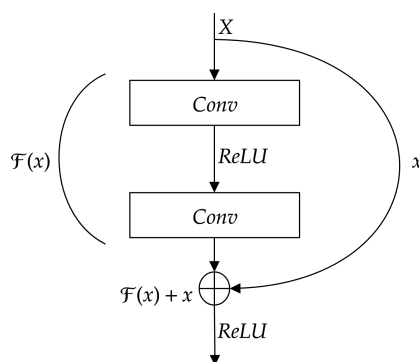
VGG Balstoties uz (Krizhevsky, Sutskever et al., 2012) iegūtajiem rezultātiem, *VGG* tīkla (Simonyan & Zisserman, 2014) autori veidoja vēl dziļāku CNN, tas ir ar vairāk slāņiem. Tas bija iespējams, jo viņi izmantoja mazus 3x3 konvolūciju kodolus, ko vēl arvien izmanto mūsdienu pētījumos. Lai nonāktu līdz beigu modeļu variantiem *VGG-19* un *VGG-16*, ar 19 un 16 slāņiem, vairākas tīklu dziļumu un citu mehānismu konfigurācijas tika plaši testētas uz *ImageNet* datu kopas. Tika atkārtoti pierādīts, ka attēlu izstiepšana un saspiešana trenēšanas laikā ļoti palīdz modeļa precizitātei, samazinot top-1 kļūdu no 27.3% uz 25.5%.

ResNet 2015. gada *ImageNet* uzdevuma uzvarētāji *ResNet* (He, X. Zhang et al., 2015) savos pētījumos novēroja, ka vienkārši liekot klāt CNN vairāk slāņus to precizitāte neuzlabojas, un pat dziļākiem tīkliem var būt lielāka kļūda nekā seklākiem.

Autori šai problēmai deva sekojošos cēloņus:

- Pazūdošo gradientu problēma, kad dziļiem tīkliem sākuma slāņu svāri tik maz ietekmē izejošās vērtības, ka kļūdas atvasinājums attiecībā pret šiem svāriem ir ļoti mazs. Tas noved pie tā, ka daļa no svāriem netiek pietiekami atjaunināti trenēšanas procesā, tomēr šo problēmu var atrisināt ar normalizāciju starp slāņiem.
- Teorijā, ja seklākam tīklam pievieno vairāk slāņus, dziļākā tīkla kļūdai nevajadzētu pārsniegt seklākā tīkla kļūdu, jo papildus slāņi var darboties kā identitātes - izdot ieejas signālu. Savukārt praksē tas ir novērojams, kas noved pie secinājuma, ka neironu tīkliem ir grūti iemācīties identitātes funkciju.

Lai varētu veidot dziļākus CNN, autori ievieš jaunu arhitektūras paveidu - dziļos paliekošos savienojumus(angļu val. *deep residual connections*), 1.8. attēls. Paraleli konvolūciju slāņiem tiek pievienota ieejas signāla identitāte, un izejas signāls ir šo divu summa. Tādā veidā gan sākuma slāņu svāriem ir lielāka ietekme uz tīkla izejas signālu, tādēļ tie tiek stiprāk modificēti atpakaļpropagācijas processā, gan vēlākie slāņi nedegradē iepriekšējo slāņu izgūtās iezīmes. *ResNet* varianti ar 34, 50 vai 101 slāņiem ir vieni no visvairāk izmantotajiem dziļajiem konvolūcijas neironu tīkliem praktiskos pielietojumos.



1.8. att. Paliekošo savienojumu shēma (veidots balstoties uz (He, X. Zhang et al., 2015))

1.3. Redzes transformatori

2017. gada pētījums (Vaswani, Shazeer et al., 2017) ieviesa jaunu dziļo neironu tīklu arhitektūru *Transformatoru*, kas pielietoja revolucionāru datu apstrādes mehānismu *uzmanību* (angļu val. *self-attention*). Šāda tipa arhitektūra un precīzāk tieši *uzmanības* mehānisms uzrādīja ievērojami labākus rezultātus vairākos NLP uzdevumos kā tulkošana, un vēlāk arī citās modalitātēs ieguva labus rezultātus. Arī datorredzes uzdevumos tiek izmantots šis mehānisms, un pēdējos gados pētnieki uzlabo un izmanto gan *uzmanību* gan konvolūciju savās arhitektūrās, jo abām pieejām ir vēlamas un nevēlamas īpašības. Šajā nodaļā aprakstīšu dziļāk *uzmanības* mehānismu un kā tas tiek pielietots datorredzes uzdevumos.

1.3.1 Uzmanības mehānisms

Uzmanības mehānisms tika veidots, lai labāk apstrādātu secīgus datus. Grūtākais uzdevums šādā domēnā ir ilgtermiņa un īstermiņa atmiņas modelēšana un iepriekš redzētas informācijas efektīva izguve balstoties uz šobrīdējo signālu. Kopš atpakaļpropagācijas izveides, tika izmantoti rekurentie neironu tīkli (balstīts uz (Rumelhart, Hinton et al., 1986)):

$$\begin{aligned} h_t &= \sigma(W_i x_t + W_h h_{t-1}) \\ y_t &= W_o h_t \end{aligned} \tag{1.11}$$

kur W_i, W_h, W_o ir ieejas, paslēptie un izejas svāri, x_t, y_t ir ieejas un izejas signāls laika solī t un h_t ir tīkla apslēptais vektors jeb atmiņa laika solī t . Ideja ir neirona tīkla iekšējo izgūto informāciju padot kā papildus ieejas signālu nākamajā solī tādā veidā izveidojot paliekošu *atmiņu*. Problēma ar šādu arhitektūru ir, ka h_t var sevī uzturēt pārāk maz informācijas par iepriekš redzētajiem signāliem.

Uzmanības pirmssākumi atrodami (Schmidhuber, 1992), kur tika piedāvāta ideja par ātrajiem un lēnajiem svāriem. Ātrie svāri, kā parasti ANN saņem ieejas signālu x_t un atgriež y_t , bet lēnie svāri saņem ieejas signālu x_t un modificē ātros, pieskaitot vai reizinot tos ar savu izejas signālu. Šādi ir iespējams likt ātro svaru tīklam koncentrēties uz svarīgo informāciju balstoties uz tajā laika solī iegūto signālu.

Pirms (Vaswani, Shazeer et al., 2017) izveidoja šobrīd zināmo *uzmanību*, bija veikti vairāki pētījumi par šo mehānismu, un jau eksistēja dažādi paveidi:

- (Bahdanau, Cho et al., 2014) izstrādā pirmo mehānismu, kas laika solī t neizmanto tikai h_{t-1} , lai iegūtu kontekstu no iepriekšējajiem ieejas signāliem, bet aplūko visus iepriekšējos h mērogojot to nozīmīgumu pēc h_t .

- (Luong, Pham et al., 2015) uzlabo mehānismu to vienkāršojot un ievieš divus uzmanības paveidus - globālo, kas aplūko visus iepriekšējos h un lokālo, kas aplūko tikai daļu. Lokālā uzmanība ir skaitļošanas ziņā lētāka, tomēr globālā parāda vislabākos rezultātus.
- (Parikh, Täckström et al., 2016) neizmanto formulu (1.11.), lai iegūtu h_t , bet izmanto parastu lineāru slāni kā $x_t \cdot W_x$. Tādējādi visi h var tikt rēķināti paralēli, kas ievērojami palielina tīkla caurlaidspēju.

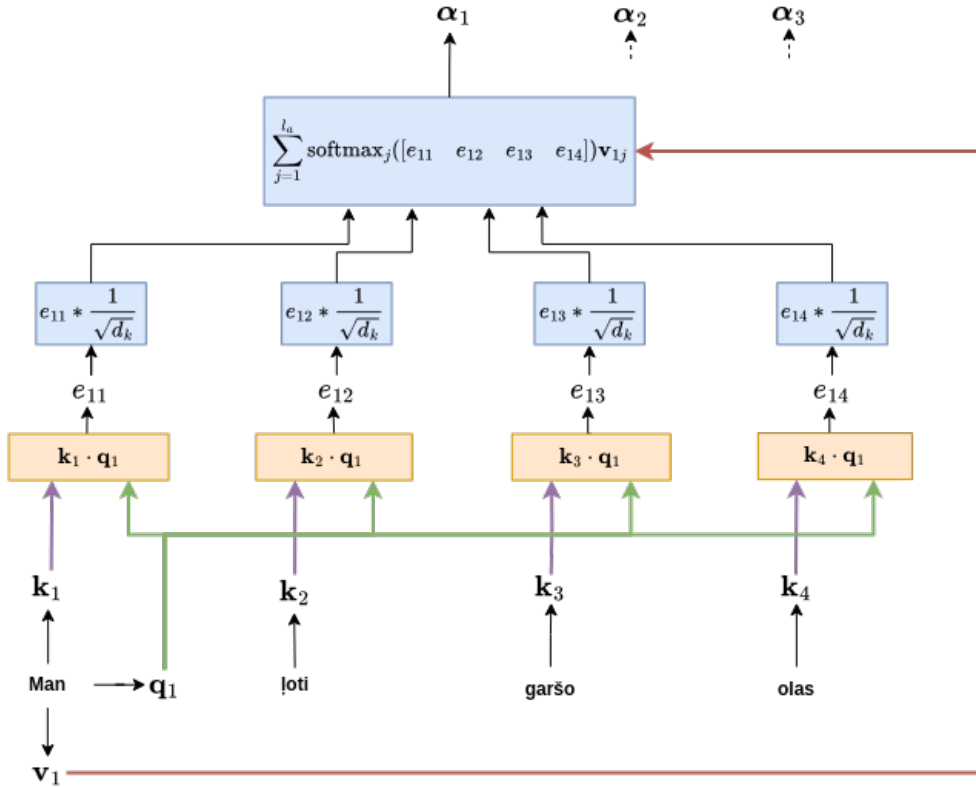
(Vaswani, Shazeer et al., 2017) ieviesa vairākus uzlabojumus iepriekš izveidotajiem mehānismiem, kā arī iepazīstināja ar kopējo tīkla arhitektūru - *Transformeru*. Matemātiski uzmanība izvērstā formā ir aprakstāma šādi (ņemts no (Vaswani, Shazeer et al., 2017)):

$$\begin{aligned}
X &= [x_1; x_2; \dots; x_T] \\
Q &= XW^Q \\
\text{softmax}(x_i) &= \frac{e^{x_i}}{\sum_j e^{x_j}} \\
\text{Attention}(Q, K, V) &= \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) \times V
\end{aligned} \tag{1.12}$$

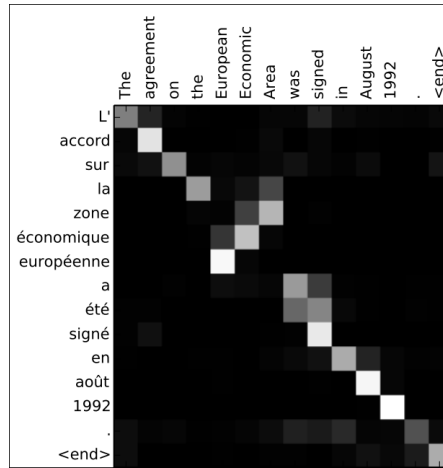
kur Q, K, V ir matricas, kas iegūtas ieejas signālus lineāri transformējot ar svāriem W^Q, W^K, W^V un d_k ir elementu skaits Q un K . Lai vieglāk izprastu uzmanības mehānismu ir jāveic dekompozīcija, un jāaplūko darbības secīgi. $W^Q \in \mathbb{R}^{L_x \times n}$, kur L_x ir vektora x garums un n ir neironu skaits, no kā izriet, ka $Q \in \mathbb{R}^{T \times n}$. Aplūkosim nevis K matricu, kas ir visu ieejas signālu lineāra transformācija, bet tikai k_t , kas ir laika solī t lineāri transformētais ieejas signāls. Reizinot k_t^T ar Q , un tad izmantojot softmax funkciju, mēs iegūstam to cik katrs ieejas elements ir svarīgs ieejas elementam x_t , jo softmax funkcijas normalizē vērtības $0 < x < 1$ un $\sum_i \text{softmax}(x_i) = 1$. Tad nozīmīguma vērtības tiek reizinātas ar V un ir iegūts uzmanības jeb adaptīvu svaru mehānisms. Intuitīvi var uz to skatīties kā:

- Q matrica (*query*) reprezentē ko katrs no ieejas signāliem meklē kā papildus informāciju.
- K matrica (*key*) vai vektors reprezentē kādu informāciju ieejas signāls piedāvā.
- V matrica (*value*) reprezentē izgūto informāciju.

Attēlā 1.9. redzama uzmanības mehānisma shēma, kur arī tiek parādīti aprēķini vektoru veidā, un attēlā 1.10. redzami nozīmīguma svaru karstuma karte starp ieejas uz izejas vārdiem tulkošanas uzdevumā.



1.9. att. Uzmanības mehānisms shēma (veidots balstoties uz (Vaswani, Shazeer et al., 2017))



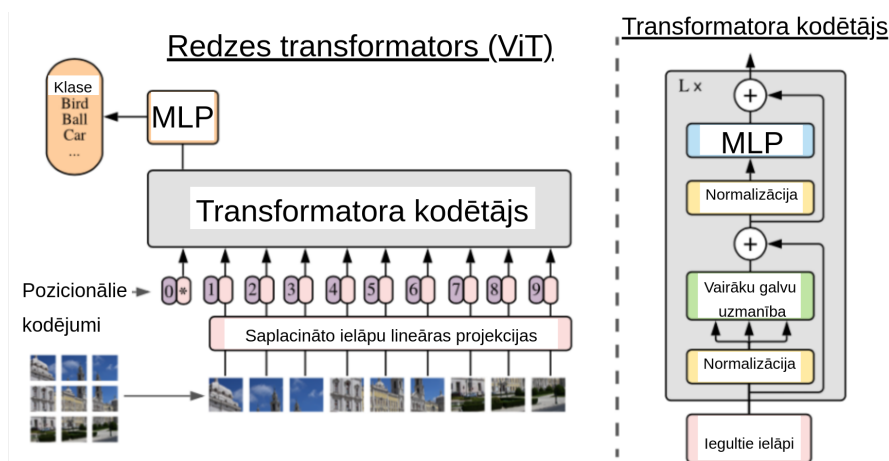
1.10. att. Uzmanības svaru vērtību/stipruma ilustrācija starp vārdiem tulkošanas uzdevumā (aizgūts no (Bahdanau, Cho et al., 2014))

1.3.2 Redzes transformatoru arhitektūra

Pēc (Vaswani, Shazeer et al., 2017) panākumiem bija vairāki mēģinājumi kā jauno atklājumu pielietot datorredzes uzdevumiem. Tomēr (Dosovitskiy, Beyer et al., 2020) bija pirmais pētījums, kur efektīvi tika pielietots uzmanības mehānisms uz attēliem. Redzes transformatorā jeb ViT nekur netika izmantotas

konvolūcijas - tikai lineārie slāņi, uzmanība un daži citi palīgmehānismi. Modelis sasniedza līdzīgus rezultātus tā brīža labākajiem konvolūciju modeļiem uz *ImageNet* datu kopas, un demonstrēja vēl labākus rezultātus uz *ImageNet-21K* datu kopas, kas sastāv no 14 miljoniem attēlu, norādot uz to, ka tā precizitāte, palielinoties datu kopas lielumam, aug labāk nekā konvolūciju tīkliem. Galvenā kritika pret šiem modeļiem ir tā ka tie ir skaitļošanas ziņā daudz dārgāki par konvolūcijas tīkliem. Ir vairāki svarīgi soļi, kas autoriem bija jāpievieno, lai veiksmīgi adaptētu uzmanību attēliem (ņemts no (Dosovitskiy, Beyer et al., 2020)):

1. Attēls tiek sadalīts vairākos ielāpos. Parasti 16x16.
2. Katrs ielāps tiek saplacināts un izlaists cauri lineāram slānim, iegūstot *iegultos ielāpus*.
3. Iegultajiem ielāpiem tiek pievienoti pozicionālie kodējumi, lai tīkls varētu labāk izmantot telpisko informāciju attēla saprašanā.
4. Visi iegultie ielāpi tiek izlaisti cauri uzmanības mehānismam. Šajā apstrādes solī modelim ir viss globālais attēla konteksts, atšķirībā no konvolūcijas kur ir tikai lokāls konteksts.
5. Izejas vektori tiek padoti uz MLP un pēc lineāra slāņa tiek minēta attēla klase.



1.11. att. Redzes transformatora arhitektūras shēma (modificēts no (Dosovitskiy, Beyer et al., 2020))

1.4. Deformējamās konvolūcijas

Deformējamās konvolūcijas ir speciāls konvolūciju paveids, kur kodols nav kvadrātveida un tā šūnas ņem ieejas signālus vienādos attālumos, bet attālumi

starp centra pikseli un šūnām ir adaptāvi un iemācāmi. Pirmo reizi ieviests (J. Dai, H. Qi et al., 2017), un vēlāk uzlabotas (Zhu, Hu et al., 2018), (W. Wang, J. Dai et al., 2022) un (Xiong, Z. Li et al., 2024), šis mehānisms tiek izmantots attēlu klasifikācijai, objektu atpazīšanai, semantiskajai segmentācijai, kā arī var tikt izmantots citos datorredzes uzdevumos. Šajā nodaļā aprakstīšu šī mehānisma pamatus un uzlabojumus, kas piedāvāti citos pētījumos.

1.4.1 Deformējamo konvolūciju pamata ideja

Deformējamās konvolūcijas jeb DCN ir salīdzinoši vienkāršs un intuitīvs mehānisms (J. Dai, H. Qi et al., 2017). Parastais konvolūcijas mehānisms sastāv no diviem soļiem (Y. LeCun, Boser et al., 1989):

1. Pikseļu ņemšana no attēla vai iezīmju kartes ar regulāru režģi;
2. Izņemto vērtību reizināšana ar svariem un saskaitīšana.

DCN 1. solis tiek aizvietots, un regulāra režģa vietā tiek izmantots deformēts režģis. Katra no parastā kodola šūnām tiek nobīdīta no savas oriģinālās pozīcijas. Nobīdes tiek izgūtas no cita konvolūcijas neironu tīkla, kam tiek padota tā pati iezīmju karte. 1.12. attēls. Aplūkotās formulas ņemtas no (J. Dai, H. Qi et al., 2017).

Piemēram, ja ir 3×3 kodols, tad ja centra pikselis atrodas vietā $(0,0)$, tad vērtības tiek ņemtas no pozīcijām:

$$\mathcal{O} = \{(-1, -1), (-1, 0), \dots, (0, 1), (1, 1)\}$$

tad parasta konvolūcija izpildītos šādi:

$$y(p) = \sum_{p_n \in \mathcal{O}} w(p_n) \cdot x(p + p_n) \quad (1.13)$$

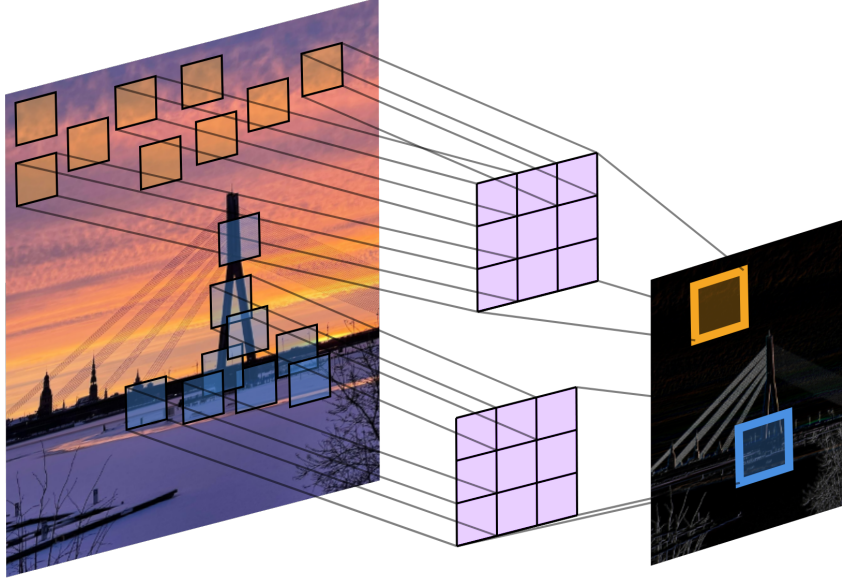
kur x ir ieejas signāls (iezīmju karte). $x(p+p_n)$ un $w(p_n)$ simbolizē vērtību ņemšanu no matricas attiecīgajā pozīcijā. Deformējamā konvolūcija:

$$y(p) = \sum_{p_n \in \mathcal{O}} w(p_n) \cdot x(p + p_n + \Delta p_n) \quad (1.14)$$

kur Δp_n ir papildus nobīdes. Nobīdes tiek izgūtas priekš visa attēla $x \in \mathbb{R}$

$$\Delta P = \text{Conv}(x) \quad (1.15)$$

kur Conv ir konvolūcijas operācija, kura saņem $x \in \mathbb{R}^{C \times H \times W}$ un atgriež nobīdes $\Delta P \in \mathbb{R}^{2N \times H \times W}$, kur $N = |\mathcal{O}|$. Nobīžu tīklam ir jāatgriež $2N$ skaitļi, tas ir x un



1.12. att. Deformējamo konvolūciju ilustrācija

y nobīde katrai kodola šūnai. Tā kā nobīdītās šūnas neatrodas perfektās pikseļu pozīcijās, tiek izmantota bilineāra interpolācija, lai iegūtu attēla vērtību ne veselā pozīcijā.

1.4.2 Deformējamo konvolūciju uzlabojumi

Pēdējos gados *DCN* ir pievienotas vairākas izmaiņas no pētījumiem, kas centušies uzlabot šo mehānismu.

Modulācija (Zhu, Hu et al., 2018) pievieno modulācijas tīklu, kas līdzīgi kā nobīžu tīkls atgriež matricu $M \in \mathbb{R}^{N \times H \times W}$, kas katrai kodola (angļu val. *kernel*) šūnai pievieno reizinātāju. Autori norāda, ka šī izmaiņa ļauj tīklam mainīt nozīmīgumu katrai no kodola šūnām, un, piemēram, ja kāda šūnai ir liela nobīde, tās reizinātājs ir mazāks, jo tā ņem attālu kontekstu.

Dziļumā atdalāmā konvolūcija (W. Wang, J. Dai et al., 2022) nomaina parastu konvolūciju uz dziļumā atdalāmo konvolūciju (Chollet, 2016), kas ir sevi pierādījusi kā variantu, kas aizņem mazāk parametrus, prasa mazāku skaitļošanu un iegūst līdzīgus rezultātus. Ja ir konvolūcijas reģions $X \in \mathbb{R}^{c_i \times 3 \times 3}$, kur c ir kanālu skaits (attēlu gadījumā 3 krāsas) un d ir telpiskās dimensijas. Parastās konvolūcijas, kuru izejas kanālu skaits ir c_o , izmanto kodolu matricu $K \in \mathbb{R}^{c_o \times c_i \times 3 \times 3}$, un katrai telpiskajai pozīcijai ir savs lineārās transformācijas vektors. Dziļumā atdalāmās konvolūcijas procesu sadala divās daļās (ņemts no (Chollet, 2016)):

1. dziļuma konvolūcija izmanto svaru matricu $K_d \in \mathbb{R}^{3 \times 3}$, un katram ieejas kanālam veic to pašu telpisko samazināšanu, un iegūst matricu $X_d \in \mathbb{R}^{c_i \times 1 \times 1}$;
2. *punktu (pointwise)* konvolūcija vai precīzāk parasta lineārā transformācija veic matricu multiplikāciju starp X_d un $K_p \in \mathbb{R}^{c_o \times c_i}$.

Grupēšana Iedvesmojoties no (Vaswani, Shazeer et al., 2017), kur tiek izmantots uzmanības mehānisms ar vairākām galvām, (W. Wang, J. Dai et al., 2022) pievieno grupēšanu. Katrai pozīcijai p , no vienādojuma (1.14.), nav tikai viens \mathcal{R} režģis, bet tiek izmantoti vairāki režģi, kuriem ir katram savas papildus izgūtās nobīdes. Tātad kādā pozīcijā p var būt vairākas deformētās konvolūcijas. Parastā konvolūcijā no šī nebūtu jēgas, jo tā ir tā pati pozīcija, bet deformējamās konvolūcijas var tad vienā pozīcijā vienlaikus koncentrēties uz dažādiem reģioniem.

Modulatora normalizācija Pēdējais uzlabojums no (W. Wang, J. Dai et al., 2022) ir modulatoru normalizācija ar *softmax* funkciju, kas parādīta (1.12.), kas ieejas vektora vērtības konvertē uz $[0,1]$ diapazonu, un vektora summa ir 1. Autori pamato, ka pārāk lielas modulatora vērtības noved pie nestabiliem gradientiem un tīkla trenēšanas procesa.

Operatora paātrinājums Vēl ir jāmin, ka (Xiong, Z. Li et al., 2024) veica pētījumus, lai uzlabotu šī mehānisma ātrumu, profilējot operatora CUDA kodu. Pētījumā tika noņemta modulatora normalizācija, un tika iegūti labāki rezultāti, tomēr apmācības process bija nestabilāks.

2. SISTEMĀTISKĀ LITERATŪRAS ANALĪZE

Bakalaura darba ietveros tika veikta sistemātiska literatūras analīze. Aplūkojām pētījumus, kur tika izstrādāti jauni dziļās mašīnmācīšanās arhitektūras veidi vai mehānismi, un kur ir iegūti labi rezultāti objektu atpazīšanas un attēlu klasifikācijas uzdevumā. Veicot analīzi tika ievākti pētījumā ieviesto modeļu rezultāti uz populārām datorredzes datu kopām, ja autori tos publicēja. Pēc pētījumu ievākšanas un dziļākas izpētes veicām kvalitatīvu un kvantitatīvu pētījumos piedāvātu sistēmu salīdzinājumu.

2.1. Pētījumu meklēšanas protokols

Pēc pirmajiem aplūkotajiem pētījumiem saistībā ar deformējamajām konvolūcijām un objektu atpazīšanas uzdevumu, secinājām, ka līdzīgi pētījumi veido attēla iezīmju izguvēju nevis koncentrējas uz objektu atpazīšanas beigu algoritmu. Tādēļ tika meklēti pētījumi, kur autori ievieš jaunu modeli, kas uzstāda labus rezultātus attēlu klasifikācijā un objektu atpazīšanā. Attēlu klasifikācija ir uzdevums, kur vieglāk salīdzināt tieši modeļa spēju izgūt informāciju no attēla. Šādi pētījumi var sniegt idejas kā labāk modificēt deformējamās konvolūcijas. Tika aplūkoti arī daži pētījumi ar specifisku fokusu uz rotācijas indispersiju vai modeļa objektu atpazīšanas *galvu* kā (Zong, G. Song et al., 2022), u.c.

Galvenie rīki, kas tika izmantoti literatūras analīzē:

- (P. contributors, 2017) var aplūkot vismodernākos paņēmienus un tabulas ar pētījumu rezultātiem uz vairāk kā 10 tūkstošiem dažādu uzdevumu. Pētījuma autori paši pievieno iegūtos rezultātus datu bāzei, tādēļ dažviet tomēr var būt nepareiza vai nepilnīga informācija par rezultātiem.
- (Wightman & P. I. M. contributors, 2019) ir dziļās mašīnmācīšanās pakotne, kur vienotā struktūrā implementēti vairāk kā 100 nozarē populāri un pēdējos gados piedāvāti datorredzes modeļi. Šeit arī ir pieejamas tabulas ar modeļu rezultātiem attēlu klasifikācijas uzdevumā uz dažādām datu kopām, izmantojot uzticamākus datus.
- (Contributors, 2023) un (K. Chen, J. Wang et al., 2019) ir Hong Kongas universitātes multimediju laboratorijas izveidotas pakotnes priekš attēlu klasifikācijas un citu datorredzes uzdevumu izpildes modeļu trenēšanai un citiem noderīgiem procesiem mašīnmācīšanās modeļu dzīves ciklā. Pieejami implementēti vairāki modeļi un to rezultāti, kā arī atbalsts trenēšanai un testēšanai uz datu kopām.

Literatūras analīzei tika pievienoti pētījumi, kuru piedāvātie modeļi ieguva augstus rezultātus uz *Imagenet*, *COCO* un *ADE20K* datu kopām un veica savus eksperimentus uz publiskām vidēja izmēra datu kopām kā *ImageNet-1K*, kā arī modeļu parametru skaits bija saglabāts tāds, lai varētu veikt objektīvu salīdzinājumu ar citiem pētījumiem. Turklāt pēc sākotnējo darbu atrašanas tika aplūkotas to references un citējumi, kā arī pievienoti pētījumi, kas bija svarīgi attiecīgā darba izstrādei vai pētījumi, kuros iegūti labi rezultāti. Tabulā 2.1. redzams literatūras analīzes apkopojums.

2.1. tabula

Sistemātiskās literatūras analīzes pētījumu saraksts

Nr	Nosaukums	Risinājuma nosaukums	Autori	Konference / Žurnāls	Datums	Citāti
1.	Deformable Convolutional Networks (J. Dai, H. Qi et al., 2017)	DCN	Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, Yichen Wei	IEEE International Conference on Computer Vision	03.2017	4241
2.	Deformable ConvNets V2: More Deformable, Better Results (Zhu, Hu et al., 2018)	DCNv2	Xizhou Zhu, Han Hu, Stephen Lin, Jifeng Dai	Computer Vision and Pattern Recognition	11.2018	1443
3.	EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (Tan & Le, 2019)	EfficientNet	Mingxing Tan, Quoc V. Le	International Conference on Machine Learning	05.2019	12875
4.	Dynamic Convolution: Attention Over Convolution Kernels (Y. Chen, X. Dai et al., 2019)	Dynamic Conv	Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, Zicheng Liu	Computer Vision and Pattern Recognition	12.2019	575
5.	End-to-End Object Detection with Transformers (Carion, Massa et al., 2020)	DETR	Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko	European Conference on Computer Vision	05.2020	8337
6.	CyCNN: A Rotation Invariant CNN using Polar Mapping and Cylindrical Convolution Layers (Kim, Jung et al., 2020)	CyCNN	Jinpyo Kim, Wookeun Jung, Hyungmo Kim, Jaejin Lee	arXiv.org	07.2020	27
7.	Deformable DETR: Deformable Transformers for End-to-End Object Detection (Zhu, Su et al., 2020)	Deformable-DETR	Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xianggang Wang, Jifeng Dai	International Conference on Learning Representations	10.2020	3120

(2.1. tabulas turpinājums)

Nr	Nosaukums	Risinājuma nosaukums	Autori	Konference / Žurnāls	Datums	Citāti
8.	An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy, Beyer et al., 2020)	ViT	Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, N. Houlsby	International Conference on Learning Representations	10.2020	19066
9.	Swin Transformer: Hierarchical Vision Transformer using Shifted Windows (Ze Liu, Y. Lin et al., 2021)	Swin	Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, B. Guo	IEEE International Conference on Computer Vision	03.2021	12025
10.	Going deeper with Image Transformers (Touvron, Cord et al., 2021)	CaiT	Hugo Touvron, M. Cord, Alexandre Sablayrolles, Gabriel Synnaeve, Herv'e J'egou	IEEE International Conference on Computer Vision	03.2021	728
11.	Scaling Local Self-Attention for Parameter Efficient Visual Backbones (Vaswani, Ramachandran et al., 2021)	HaloNet	Ashish Vaswani, Prajit Ramachandran, A. Srinivas, Niki Parmar, Blake A. Hechtman, Jonathon Shlens	Computer Vision and Pattern Recognition	03.2021	298
12.	Emerging Properties in Self-Supervised Vision Transformers (Caron, Touvron et al., 2021)	DINO-pretrain	Mathilde Caron, Hugo Touvron, Ishan Misra, Herv'e J'egou, J. Mairal, Piotr Bojanowski, Armand Joulin	IEEE International Conference on Computer Vision	04.2021	3219
13.	VOLO: Vision Outlooker for Visual Recognition (Yuan, Hou et al., 2021)	VOLO	Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, Shuicheng Yan	IEEE Transactions on Pattern Analysis and Machine Intelligence	06.2021	223
14.	XCiT: Cross-Covariance Image Transformers (El-Nouby, Touvron et al., 2021)	XCiT	Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, I. Laptev, N. Neverova, Gabriel Synnaeve, Jakob Verbeek, H. Jégou	Neural Information Processing Systems	06.2021	352
15.	Refiner: Refining Self-attention for Vision Transformers (D. Zhou, Y. Shi et al., 2021)	Refiner	Daquan Zhou, Yujun Shi, Bingyi Kang, Weihao Yu, Zihang Jiang, Yuan Li, Xiaojie Jin, Qibin Hou, Jiashi Feng	arXiv.org	06.2021	43

(2.1. tabulas turpinājums)

Nr	Nosaukums	Risinājuma nosaukums	Autori	Konference / Žurnāls	Datums	Citāti
16.	CoAtNet: Marrying Convolution and Attention for All Data Sizes (Z. Dai, H. Liu et al., 2021)	CoAtNet	Zihang Dai, Hanxiao Liu, Quoc V. Le, Mingxing Tan	Neural Information Processing Systems	06.2021	822
17.	MViTv2: Improved Multiscale Vision Transformers for Classification and Detection (Y. Li, C. Wu et al., 2021)	MViTv2	Yanghao Li, Chaoxia Wu, Haoqi Fan, K. Mangalam, Bo Xiong, J. Malik, Christoph Feichtenhofer	Computer Vision and Pattern Recognition	12.2021	419
18.	Masked-attention Mask Transformer for Universal Image Segmentation (Cheng, Misra et al., 2021)	Mask2Former	Bowen Cheng, Ishan Misra, A. Schwing, Alexander Kirillov, Rohit Girdhar	Computer Vision and Pattern Recognition	12.2021	1073
19.	Vision Transformer with Deformable Attention (Xia, Pan, S. Song et al., 2022)	DAT	Zhuofan Xia, Xuran Pan, S. Song, Li Erran Li, Gao Huang	Computer Vision and Pattern Recognition	01.2022	218
20.	A ConvNet for the 2020s (Zhuang Liu, Mao et al., 2022)	ConvNext	Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, Saining Xie	Computer Vision and Pattern Recognition	01.2022	2678
21.	Focal Modulation Networks (Yang, C. Li et al., 2022)	FocalNet	Jianwei Yang, Chunyuan Li, Jianfeng Gao	Neural Information Processing Systems	03.2022	117
22.	DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection (H. Zhang, F. Li et al., 2022)	DINO-DETR	Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun-Juan Zhu, L. Ni, H. Shum	International Conference on Learning Representations	03.2022	600
23.	DaViT: Dual Attention Vision Transformers (M. Ding, B. Xiao et al., 2022)	DaViT	Mingyu Ding, Bin Xiao, N. Codella, P. Luo, Jingdong Wang, Lu Yuan	European Conference on Computer Vision	04.2022	126
24.	MaxViT: Multi-Axis Vision Transformer (Tu, Talebi et al., 2022)	MaxViT	Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, P. Milanfar, A. Bovik, Yinxiao Li	European Conference on Computer Vision	04.2022	304
25.	Global Context Vision Transformers (Hatamizadeh, Yin et al., 2022)	GCViT	Ali Hatamizadeh, Hongxu Yin, J. Kautz, Pavlo Molchanov	International Conference on Machine Learning	06.2022	53
26.	HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions (Rao, W. Zhao et al., 2022)	HorNet	Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, S. Lim, Jiwen Lu	Neural Information Processing Systems	07.2022	136

(2.1. tabulas turpinājums)

Nr	Nosaukums	Risinājuma nosaukums	Autori	Konference / Žurnāls	Datums	Citāti
27.	More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity (S. Liu, T. Chen et al., 2022)	SLaK	Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Q. Xiao, Boqian Wu, Mykola Pechenizkiy, D. Mocanu, Zhangyang Wang	International Conference on Learning Representations	07.2022	92
28.	UniNet: Unified Architecture Search with Convolution, Transformer, and MLP (J. Liu, H. Li et al., 2021)	UniNet	Jihao Liu, Hongsheng Li, Guanglu Song, Xin Huang, Yu Liu	European Conference on Computer Vision	07.2022	21
29.	Dilated Neighborhood Attention Transformer (Hassani & H. Shi, 2022)	DiNAT	Ali Hassani, Humphrey Shi	arXiv.org	09.2022	33
30.	MetaFormer Baselines for Vision (Yu, Si et al., 2022)	CAFormer	Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, Xinchao Wang	IEEE Transactions on Pattern Analysis and Machine Intelligence	10.2022	45
31.	DETRs with Collaborative Hybrid Assignments Training (Zong, G. Song et al., 2022)	Co-DETR	Zhuofan Zong, Guanglu Song, Yu Liu	IEEE International Conference on Computer Vision	11.2022	73
32.	RIC-CNN: Rotation-Invariant Coordinate Convolutional Neural Network (Mo & G. Zhao, 2022)	RIC-CNN	Hanlin Mo, Guoying Zhao	Pattern Recognition	11.2022	5
33.	InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions (W. Wang, J. Dai et al., 2022)	InternImage	Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiao-hua Hu, Tong Lu, Lewei Lu, Hongsheng Li, Xiaogang Wang, Y. Qiao	Computer Vision and Pattern Recognition	11.2022	266
34.	Reversible Column Networks (Y. Cai, Y. Zhou et al., 2022)	RevCol	Yuxuan Cai, Yi Zhou, Qi Han, Jianjian Sun, Xiangwen Kong, Jun Yu Li, Xiangyu Zhang	International Conference on Learning Representations	12.2022	18
35.	ONE-PEACE: Exploring One General Representation Model Toward Unlimited Modalities (P. Wang, S. Wang et al., 2023)	One-Peace	Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, Chang Zhou	arXiv.org	05.2023	35
36.	Scale-Aware Modulation Meet Transformer (W.-S. Lin, Z. Wu et al., 2023)	SMT	Wei-Shiang Lin, Ziheng Wu, Jiayu Chen, Jun Huang, Lianwen Jin	IEEE International Conference on Computer Vision	07.2023	11

(2.1. tabulas turpinājums)

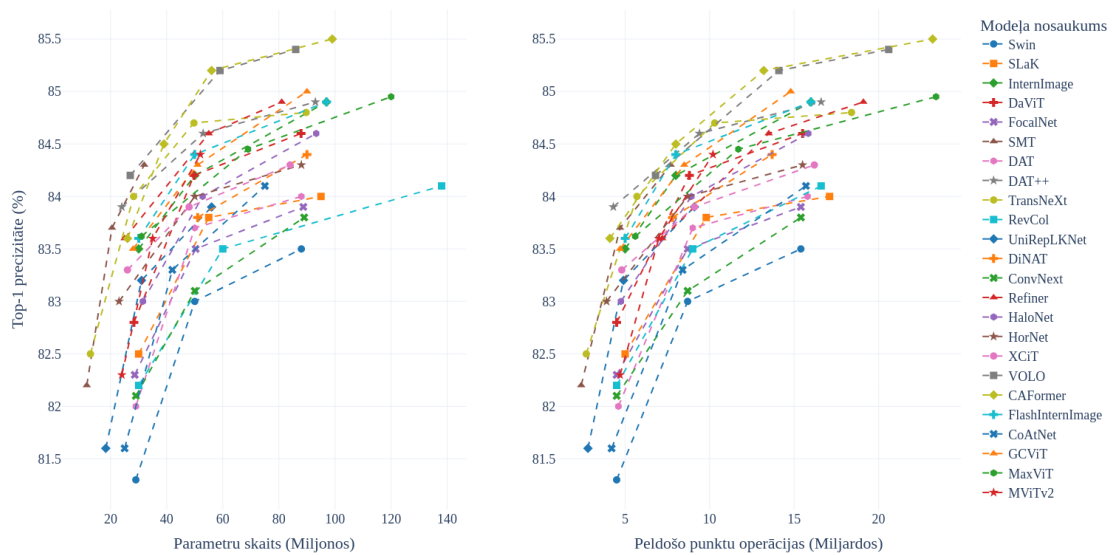
Nr	Nosaukums	Risinājuma nosaukums	Autori	Konference / Žurnāls	Datums	Citāti
37.	DAT++: Spatially Dynamic Vision Transformer with Deformable Attention (Xia, Pan, Shiji Song et al., 2023)	DAT++	Zhuofan Xia, Xuran Pan, Shiji Song, Li Er-ran Li, Gao Huang	arXiv.org	09.2023	2
38.	TransNeXt: Robust Foveal Visual Perception for Vision Transformers (D. Shi, 2023)	TransNeXt	Dai Shi	arXiv.org	11.2023	1
39.	UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition (X. Ding, Y. Zhang et al., 2023)	UniRepLKNet	Xiaohan Ding, Yiyuan Zhang, Yixiao Ge, Sijie Zhao, Lin Song, Xiangyu Yue, Ying Shan	arXiv.org	11.2023	4
40.	Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications (Xiong, Z. Li et al., 2024)	Flash-InternImage	Yuwen Xiong, Zhiqi Li, Yuntao Chen, Feng Wang, Xizhou Zhu, Jiapeng Luo, Wenhai Wang, Tong Lu, Hongsheng Li, Yu Qiao, Lewei Lu, Jie Zhou, Jifeng Dai	arXiv.org	01.2024	1

2.2. Kvantitatīvs salīdzinājums

Šajā nodaļā aplūkosim ievāktu pētījumu publicētos rezultātus uz attēlu klasifikācijas, objektu atpazīšanas un semantiskās segmentācijas uzdevumiem. Pētījumos tiek novērtēti dažāda izmēra modeļi, kas iegūst mazāku precizitāti, paturot zemāku parametru skaitu. Aplūkotie rādītāji aprakstīti 3.2. nodaļā, bet datu kopas 3.1. nodaļā.

2.1. attēlā redzami rezultāti attēlu klasifikācijas uzdevumā. Kā var redzēt vairāki aspekti ietekmē modeļa precizitāti un pat, salīdzinot precizitāti ar modeļa izmēru un kompleksitāti, nevar pateikt vai precizitātes uzlabojums patiešām nāk no mehānisma spējas labāk procesēt ieejas signālu.

2.2. attēlā ir rezultāti uz COCO datu kopas. Pētījumos ieviestie modeļi izmanto divas dažādas detekcijas galvas arhitektūras. No abām *Cascade Mask R-CNN* ir labāka, tādēļ pētījumi tiek salīdzināti atsevišķi. Var novērot, ka objektu detekcijas rezultāti korelējas ar attēlu klasifikāciju, tomēr ir dažas izmaiņas.

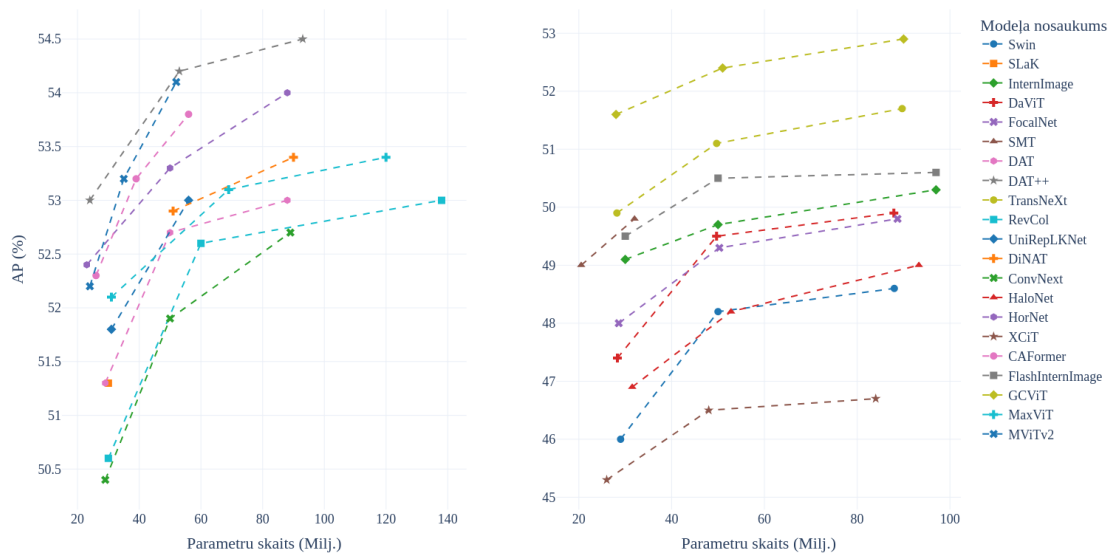


2.1. att. Modeļu salīdzinājums uz ILSVRC12 (Deng, Dong et al., 2009) validācijas kopas

2.3. Kvalitatīvs salīdzinājums

Pēc literatūras izpētes atlasītie pētījumi tika filtrēti, lai varētu veikt objektīvu salīdzinājumu starp piedāvātajiem modeļiem. Mērķis ir salīdzināt tieši pētījumos izveidotus modeļus, kas ir paredzēti vairākiem datorredzes uzdevumiem - pamatā klasifikācijai, detekcijai un segmentācijai. Ir svarīgi arī tas vai modeļa pirmkods un uztrenētie svāri ir pieejami, lai varētu atkārtot novērtējumu un veikt dziļāku iegūto rezultātu izpēti, kā arī risinājumu salīdzinājumu uz atsevišķiem piemēriem. Kvalitatīvais salīdzinājums redzams tabulā 2.2.. Tika izmantoti sekojošie kritēriji:

- K1 Pētījuma galvenais pienesums ir jauns datorredzes modelis, kas var tikt izmantots attēlu klasifikācijai un objektu atpazīšanai;
- K2 Ir brīvi pieejams modeļa arhitektūras un mehānismu pirmkods;
- K3 Ir brīvi pieejami eksperimentu rezultātā iegūtie modeļu svāri;
- K4 Pētījumā ir publicēti modeļa novērtējums uz *ImageNet-1K* validācijas datu kopas un *COCO-2017* validācijas kopas;
- K5 Tas kādu pamata mehānismu modelis izmanto, norādīts ar burtiem:
 - C konvolūcija;
 - A uzmanība;
 - H hibrīds;



2.2. att. Modeļu salīdzinājums uz COCO (T.-Y. Lin, Maire et al., 2014) validācijas kopas. Kreisajā pusē ar *Cascade Mask R-CNN* (Z. Cai & Vasconcelos, 2019) detekcijas galvu, labajā *Mask R-CNN* (He, Gkioxari et al., 2017)

X cits mehānisms;

- nav piemērojams.

Ar * norādām, ka attiecīgais mehānisms nav standarta un ir kaut kādā veidā modificēts.

2.2. tabula

Kvalitatīvs literatūras analīzes pētījumu salīdzinājums

Nr	Risinājuma nosaukums	K1	K2	K3	K4	K5
1.	DCN	✓	✓	✓	✓	C*
2.	DCNv2	✓	✓	✓	✓	C*
3.	EfficientNet	✓	✓	✓	✓	C
4.	Dynamic Conv		✓	✓	✓	H
5.	DETR		✓	✓		A
6.	CyCNN		✓	✓		C
7.	Deformable-DETR		✓	✓		-
8.	ViT		✓	✓	✓	A
9.	Swin	✓	✓	✓	✓	A
10.	CaiT	✓	✓	✓	✓	A
11.	HaloNet	✓	✓	✓	✓	A
12.	DINO-pretrain		✓			-

(2.1. tabulas nobeigums)

Nr	Risinājuma nosakums	K1	K2	K3	K4	K5
13.	VOLO	✓	✓	✓	✓	A*
14.	XCiT	✓	✓	✓	✓	A*
15.	Refiner		✓	✓	✓	A*
16.	CoAtNet	✓	✓	✓	✓	H
17.	MViTv2	✓	✓	✓	✓	A*
18.	Mask2Former		✓	✓		X
19.	DAT	✓	✓	✓	✓	A
20.	ConvNext	✓	✓	✓	✓	C
21.	FocalNet	✓	✓	✓	✓	X
22.	DINO-DETR		✓	✓	✓	-
23.	DaViT	✓	✓	✓	✓	A*
24.	MaxViT	✓	✓	✓	✓	A*
25.	GCViT	✓	✓	✓	✓	A*
26.	HorNet	✓	✓	✓	✓	C*
27.	SLaK	✓	✓	✓	✓	C
28.	UniNet	✓	✓	✓	✓	H
29.	DiNAT	✓	✓	✓	✓	A*
30.	CAFormer	✓	✓	✓	✓	H
31.	Co-DETR		✓	✓		-
32.	RIC-CNN		✓	✓		C*
33.	InternImage	✓	✓	✓	✓	C*
34.	RevCol		✓	✓	✓	X
35.	One-Peace		✓	✓		H
36.	SMT	✓	✓	✓	✓	H
37.	DAT++	✓	✓	✓	✓	A*
38.	TransNeXt	✓	✓		✓	A*
39.	UniRepLKNet	✓	✓	✓	✓	C*
40.	FlashInternImage	✓	✓	✓	✓	C*

3. METODOLOĢIJA

Šajā nodaļā tiek aprakstītas pētījumā izmantotās datu kopas un izvēlētie veikspējas rādītāji. Tāpat tiek aprakstītas arī eksperimentos aplūkoto modeļu arhitektūras un to apmācības un testēšanas protokols.

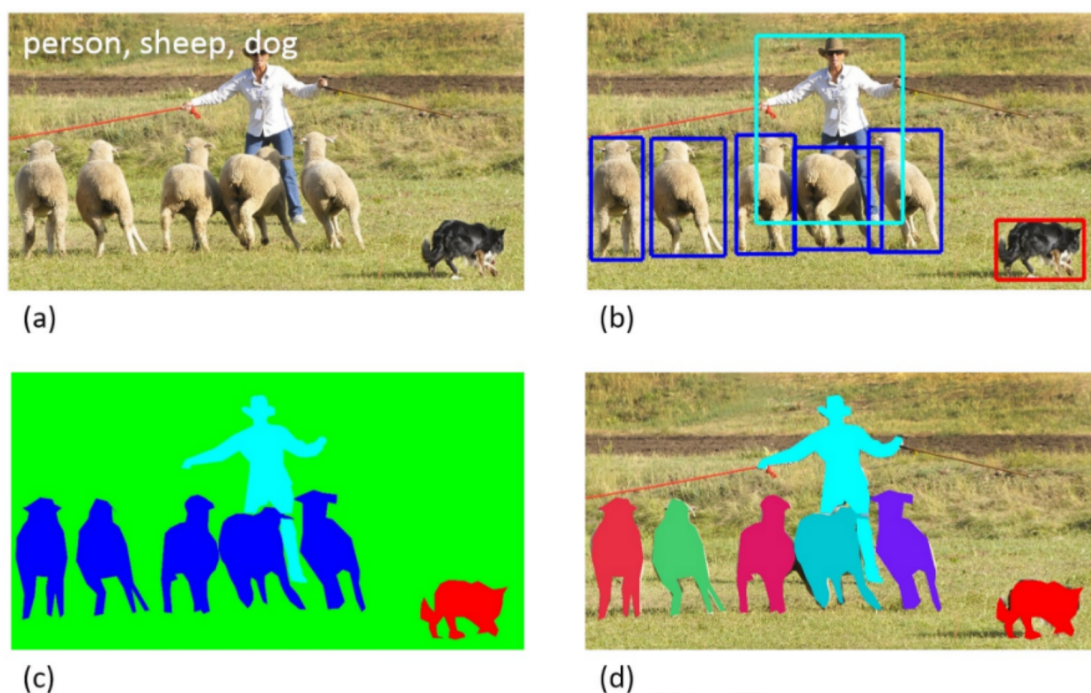
3.1. Datu kopas

Veicot sistemātisko literatūras analīzi, tika secināts, ka gandrīz visi aplūkotie pētījumi, pirms modeļa apmācības objektu atpazīšanā, tas tiek apmācīts attēlu klasifikācijas uzdevumā. Ņemot šo vērā, tiks aplūkotas gan attēlu klasifikācijas datu kopas, gan objektu atpazīšanas datu kopas.

ILSVRC12 ILSVRC12 datu kopa ir daļa no populārās ImageNet (Deng, Dong et al., 2009) datu kopas, kas ir balstīta uz *Wordnet* (Fellbaum, 1998) vārdu hierarhijas. Tās uzdevums ir attēlu klasifikācija pa 1000 dažādām klasēm, kas ir lapas vai iekšējie mezgli, bet nepārklājas, *Wordnet* vārdu hierarhijā. Katrai klasei ir 50 validācijas attēli, 100 testa attēli un ap, bet ne vairāk kā, 1300 trenēšanas attēli. Lai iegūtu trenēšanas un validācijas attēlus, oficiālajā ImageNet mājaslapā tos vajadzēja pieprasīt, izmantojot augstskolas e-pastu. Testēšanas attēli ir pieejami, bet to klašu marķējumi nav publiski pieejami, tomēr ir iespējams novērtēt modeli, sūtot tā minējumus uz *ImageNet* serveri. Ir noderīgi minēt, ka daudzi pētnieki uzskata, ka tieši ImageNet datu kopa ir visvairāk atbildīga par mākslīgā intelekta nozares uzplaukumu pēdējos gados, sniedzot pētniekiem labu datu kopu, kur izstrādāt algoritmus, un tos salīdzināt ar citu izstrādātajiem risinājumiem.

MS COCO *Microsoft* bieži sastopami objekti kontekstā (*common objects in context*) ir visizmantotākā objektu atpazīšanas datu kopa. Tā sastāv no 328 tūkstošiem attēlu, 2.8 miljoniem nomarkētu objektu ģeogrāfisko izgriezumumu un instances masku katram objektam starp 91 kategoriju. (T.-Y. Lin, Maire et al., 2014) ieviestā datu kopa tika atjaunināta 2017. gadā. Tās trenēšanas kopa sastāv no 118 tūkstošiem attēlu, validācijas 5 tūkstoši un 41 tūkstošiem testēšanas attēlu. Trenēšanas un validācijas kopas ir publiski pieejamas, bet lai iegūtu novērtējumu uz testēšanas kopu, rezultāti ir jāsūta datu kopas veidotājiem.

Small-ImageNet *Small-ImageNet* ir mūsu ieviesta datu kopa, kas ir apakškopa no ILSVRC12. Ņemot vērā ILSVRC12 lielo izmēru un skaitļošanas resursu pieejamību, tika izveidota mazāka datu kopa, uz ko var vieglāk veikt ablācijas pētījumus ar modifikācijām. Klašu skaits samazinās no 1000 uz 200, un tās tiek



3.1. att. (a) Attēlu klasifikācijas. (b) Objektu lokalizācija. (c) Semantiskā segmentācija. (d) Objektu atpazīšana, MS COCO datu kopas uzdevums (modificēts no (T.-Y. Lin, Maire et al., 2014))

ņemtas no (mnmostafa, 2017). Katrai no klasēm pēc nejaušības principa tiek atstāti tikai 200 attēli priekš apmācības, un validācijas attēlu skaits paliek tāds pats. Datu kopas izveides kods ir atrodams bakalaura darba atvērtajā pirmkodā un nejaušība tiek implementēta tā, lai datu kopa būtu tāda pati.

COCO minitrain (Samet, Hicsonmez et al., 2020) ievieš arī samazinātu apakškopu COCO datu kopai. Attēli tika izvēlēti tā, lai samazināt objektu kategoriju distribūciju. Autori ar eksperimentiem parāda, ka rezultāti uz mazās apakškopas korelē ar veikspēju uz lielā COCO.

3.2. Rādītāji

Šajā nodaļā aprakstīšu rādītājus, kas tiks izmantoti, lai salīdzinātu pētījumos ieviestos risinājumus attiecīgajos uzdevumos. Sākotnēji katrā no uzdevumiem rādītājus izveidoja etalonuzdevumu veidotāji kā (Deng, Dong et al., 2009), bet vēlāk parādījās citi rādītāji, ko ieviesa kopiena vai, kas bija noderīgi produkcijas vidē.

3.2.1 Klasifikācija

Šajā uzdevumā modelis katram attēlam atgriež sarakstu ar varbūtībām, kur katra varbūtība atbilst vienai no klasēm kas ir jāmin. Varbūtību summa ir 1, un lai iegūtu modeļa minējumu ir jāizvēlas klase, kura atbilst vislielākajai varbūtībai. Rādītāji iegūti no (Bishop, 2007) un (Deng, Dong et al., 2009).

Precizitāte (*Accuracy*) Šis rādītājs novērtē, cik daudz procentuāli modelis ir pareizi uzminējis no piemēriem datu kopā.

$$\text{Precizitāte} = \frac{1}{N} \sum_{i=1}^N 1(y_i = \hat{y}_i) \quad (3.1)$$

kur y ir klases minējumi un \hat{y} ir īstās attēlu etiķetes. Šie ir reāli skaitļi un katrs skaitlis atbilst klasei.

Top- k precizitāte *ImageNet* etalonuzdevumā tiek izmantoti rādītāji kā top-1 precizitāte un top-5 precizitāte. Precizitāte formulā (3.1.) ir viens no gadījumiem top- k precizitātei, kad $k = 1$. Universāla formula izskatītos šādi:

$$\text{Top-}k \text{ precizitāte} = \frac{1}{N} \sum_{i=1}^N 1(\hat{y}_i \in Y_{i,k}) \quad (3.2)$$

kur $Y_{i,k}$ ir modeļa k augstākās pārliecības minējumi i attēlam. Piemēram, ja $k = 5$, tad pareizi uzminēts piemērs tiek ieskaitīts ja pareizā klase ir iekš vienas no piecām augstākās pārliecības minētājām klasēm.

Pārklājums (*Recall*) Bieži precizitāte nav pietiekama metrika, lai noteiktu cik labs ir modelis un vai to var izmantot īstā dzīves uzdevumā. Piemēram, ja ir kāda klase, ko ir ļoti svarīgi uzminēt, tad pārklājums ir rādītājs, ko izmantot. Vienkārši izsakoties tā ir attiecība starp to, cik modelis uzminēja un cik bija jāuzmin šajā klasē kopā. Parasti pārklājuma rādītāju aplūko katrai klasei atsevišķi:

$$\text{Pārklājums} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.3)$$

kur TP ir pareizi uzminētie piemēri attiecīgajai klasei un FN ir kļūdaini negatīvie - piemēri, kur pareizā atbilde bija attiecīgā klase, bet modelis paredzēja kādu citu klasi.

Precīzija (*Precision*) Precīzija ir pārklājuma otra puse - attiecība starp to, cik modelis uzminēja pareizi attiecīgo klasi un cik kopā modelis minēja šo klasi.

$$\text{Precīzija} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3.4)$$

kur FP ir piemēri, kur modelis minēja attiecīgo klasi, bet pareizā atbilde bija cita. Precīzijai un pārklājumam arī ir iespējams izrēķināt to top- k variantus, bet šos parasti neizmanto.

3.2.2 Semantiskā segmentācija

Semantiskajā segmentācijā līdzīgi kā klasifikācijā tiek minētas klašu varbūtības kādam attēlam tikai tas tiek darīts katram attēla pikselim. Klašu varbūtības katram pikselim tiek konvertētas uz minēto klasi katram pikselim, ņemot augstāku varbūtību. Pikseli ar marķējuma klases vērtību 0 apzīmē, ka tur nekas neatrodas - fona pikselis.

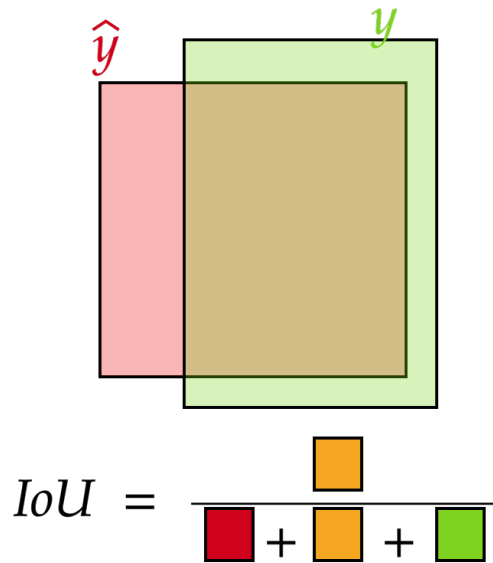
Šķēlums pāri apvienojumam (*IoU*) Šķēlums pāri apvienojumam jeb *IoU* vai *Jaccard index* ir divu kopu šķēluma lielums dalīts ar to apvienojumu:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.5)$$

Šajā gadījumā kopas A un B būtu modeļa klases minējums katram pikselim un īstā klase katram pikselim. Saprotamāka formula attēliem būtu:

$$\begin{aligned} I &= \sum_{i=0}^h \sum_{j=0}^w 1(Y_{i,j} = \hat{Y}_{i,j}) \\ U &= \sum_{i=0}^h \sum_{j=0}^w 1(Y_{i,j} \neq 0 \mid \hat{Y}_{i,j} \neq 0) \\ \text{IoU} &= \frac{I}{U} \end{aligned} \quad (3.6)$$

kur $Y \in \mathbb{Z}^{h \times w}$ ir modeļa minējumu matrica katram pikselim, $\hat{Y} \in \mathbb{Z}^{h \times w}$ ir īsto klašu matrica un h un w ir attēla augstums un garums. Šis rādītājs ir ļoti līdzīgs klasifikācijas precizitātei tikai priekš segmentācijas. Tas precīzi tika noformulēts (Everingham, Gool et al., 2010). 3.2. attēlā ir redzama šķēluma pāri apvienojumam rādītāja jeb *IoU* ilustrācija.



3.2. att. Rādītāja šķelums pāri apvienojumam ilustrācija

3.2.3 Objektu atpazīšana

Objektu atpazīšanā ne tikai ir jāmin katra pikseļa klase, bet arī tas, kurai instancei šis objekts pieder. Bez objekta maskas ir jāmin objekta ģeometriskais ietvars, skatīt 3.1. attēlu. Ja augstāk minētajos uzdevumos modeļa minējums un īstais marķējums ir viendabīgs, tad objektu atpazīšanā tā nav, jo minējumu skaits un īsto objektu skaits var būt atšķirīgs. Šis fakts sarežģa rādītāju aprēķinus. Katrs no modeļa minējumiem sastāv no:

klases - kategorijas identifikatora;

pārlicības - skaitli robežās $[0,1]$;

ģeometriskā reģiona - x un y koordinātas augšējam kreisajam stūrim un reģiona augstums un platums;

maskas - ne obligāti objekta maska, tas ir pikseļu indeksu kopas, kas pieder šai instancei.

Vidējā precīzija (AP) Vidējā precīzija ir galvenais rādītājs objektu atpazīšanas uzdevumā. To aprēķina sekojošajos soļos:

1. Nodēfinē IoU sliekšni un pārlicības sliekšni, piemēram 0.5 abiem.
2. Visus minējumus zem pārlicības sliekšņa atmet.

3. Iterē cauri visiem marķētajiem, īstajiem objektiem un aprēķina IoU ar katru no modeļa minējumiem ar tādu pašu klasi. Izmanto formulu (3.5.), kopu A un B elementi ir pikseli iekš minējuma un marķējuma reģioniem attiecīgi.
4. Ja ir kādi minējumi virs IoU sliekšņa, tad minējums ar vislielāko IoU tiek atzīmēts kā pareizs un izņemts no pārējiem aprēķiniem, kā arī marķētais objekts tiek atzīmēts kā uzminēts.

Tagad ir iegūti tādi paši rādītāji kā klasifikācijas uzdevumā:

TP pareizo minējumu skaits vai uzminēto marķējumu skaits;

FN neuzminēto marķējumu skaits;

FP minējumu skaits, kas nav pareizi;

TN objektu atpazīšanu uzdevumā šo nevar nodefinēt.

Var izmantot šos rādītājus, lai aprēķinātu rādītājus formulās (3.3.) un (3.4.). Tomēr AP rādītājs ir vēl sarežģītāks:

$$\text{Vidējā Precīzija} = \int_{r=0}^1 p(r) dr \quad (3.7)$$

kur $p(r)$ ir precīzija, kad pārlicības sliekšnis ir tāds, ka pārklājums $= r$ izmantojot formulas (3.3.) un (3.4.) un augstāk minēto algoritmu. Pie maza pārlicības sliekšņa precīzitāte ir augsta, jo ir maz FP, bet augsts pārklājums, bet pie augsta pārlicības sliekšņa ir otrādāk. Vidējās precīzijas rādītājs aprēķina laukuma zem šīs attiecības līknes. Modeļu salīdzinājumos šie ir trīs izmantotākie AP varianti:

AP₅₀ vidējā precīzija, kad IoU sliekšnis ir 0.5;

AP₇₅ vidējā precīzija, kad IoU sliekšnis ir 0.75;

AP_[.5:.05:.95] vai vienkārši AP ir vidējais starp vidējām precīzijām pie IoU sliekšņa $\in \{0.5, 0.55, \dots, 0.9, 0.95\}$. Šis ir galvenais rādītājs objektu atpazīšanas uzdevumā.

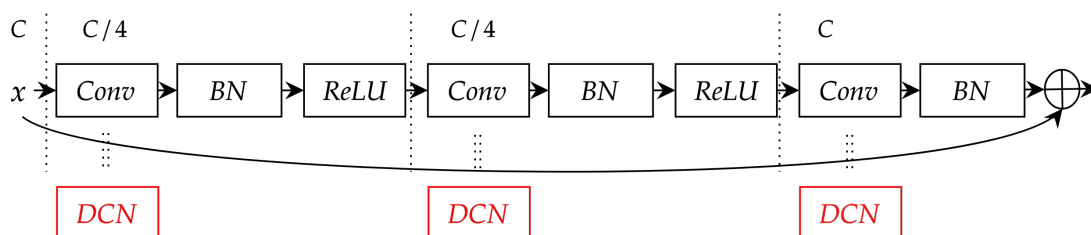
Vidējā precīzija maskām (*AP_{mask}*) Vidējā precīzija maskām ir gandrīz tāda pati kā parastā, tikai, lai aprēķinātu IoU starp minējumu un īsto marķējumu tiek izmantota formula (3.6.) un minējuma maska. Tiek izmantoti arī tādi paši rādītāja varianti.

3.3. Pētīto modeļu arhitektūras

Bakalaura darba praktiskās daļas ietvaros tika izveidotas un apmācītas trīs dažādas dziļo neironu tīklu arhitektūras. Divas no tām izmanto jau iepriekš izveidotus mehānismus, un salīdzina to veikspēju pret bieži izmantoto attēlu apstrādes neironu tīklu *ResNet* (He, X. Zhang et al., 2015). Trešā arhitektūra izmanto jaunieviestu mehānismu *LightDCN*, kas ir DCN modifikācija.

3.3.1 Deformējamās konvolūcijas iekš *ResNet*

ResNet (He, X. Zhang et al., 2015) arhitektūra sastāv no 4 slāņiem, kuri katrs sastāv no vairākiem blokiem. Šai arhitektūrai ir vairāki varianti, kur tiek palielināts bloku skaits, lai uzlabotu precizitāti, palielinot skaitļošanas sarežģītību. Kā pamata arhitektūras variants tika izvēlēts *resnet50*, kas izmanto sašaurinājuma (angļu val. *bottleneck*) bloku, kopā sastāv no 50 konvolūcijas slāņiem un tam ir ~ 25.6 miljoni apmācāmu parametru. Šis variants tika izvēlēts, jo tas ir viens no biežāk izmantotajiem neironu tīkliem nozarē. Tā parametru skaits ir apmēram tāds pats kā deformējamo konvolūciju modeļa mazāko variantu un to var apmācīt pietiekami ātri ar mums pieejamajiem skaitļošanas resursiem. Attēlā 3.3. redzama shēma, kas parāda kurās vietās tiek ievietotas deformējamās konvolūcijas. Bloki kā BN (Ioffe & Szegedy, 2015) un *ReLU* (1.1.3. nodaļa) aktivizācijas funkcija tiek atstāti, un izmainīta tikai konvolūcija pret deformējamo konvolūciju. Tika apmācītas 5 dažādas arhitektūras, parastā *ResNet* un 4 paveidi, kur tiek aizstāti pēdējo slāņu bloki pret deformējamo konvolūciju pudeles kakla blokiem.

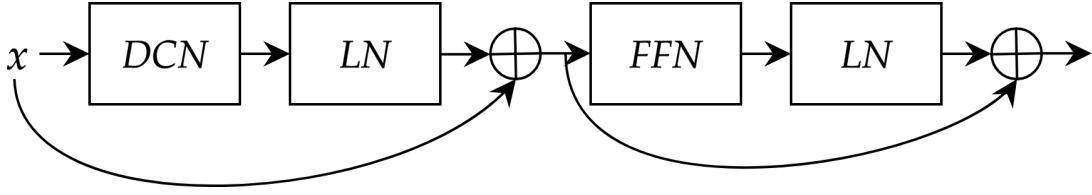


3.3. att. Sašaurinājuma bloka un sašaurinājuma ar DCN shēma.
Pamata arhitektūra ir iegūta no (He, X. Zhang et al., 2015)

3.3.2 *InternImage* bloki iekš *Resnet*

(W. Wang, J. Dai et al., 2022) ir nesnāka pētījums, kas izmanto deformējamās konvolūcijas iekš savas piedāvātās arhitektūras - *InternImage*, un kombinē šo mehānismu ar citiem moderniem blokiem, kas izmantoti ViT modeļos, kā LN un *GELU* (Hendrycks & Gimpel, 2016), kā arī FFN un *Transformatora*

arhitektūru (Vaswani, Shazeer et al., 2017). Attēlā 3.4. redzama bloka shēma. Tika apmācītās 4 arhitektūras - parastā *ResNet* arhitektūra, pēdējā slāņa bloku aizstāšana ar *InternImage* blokiem, pirmā un pirmo divu slāņu bloku aizstāšana. Aizstājot vairāk par vienu pēdējo slāni ievērojami palielinās patērētais GPU atmiņas daudzums, un ar pieejamajiem resursiem nebūtu iespējams veikt līdzvērtīgus apmācības eksperimentus. Vietās, kur nesakrīt *ResNet* un *InternImage* kanālu skaita palielinājums (*ResNet* pirmajā slānī kanālu skaitu palielina 4 reizes, bet *InternImage* visur tikai 2) tiek pievienots Conv slānis ar kodola izmēru 1 un palielina kanālu skaitu 2 reizes.

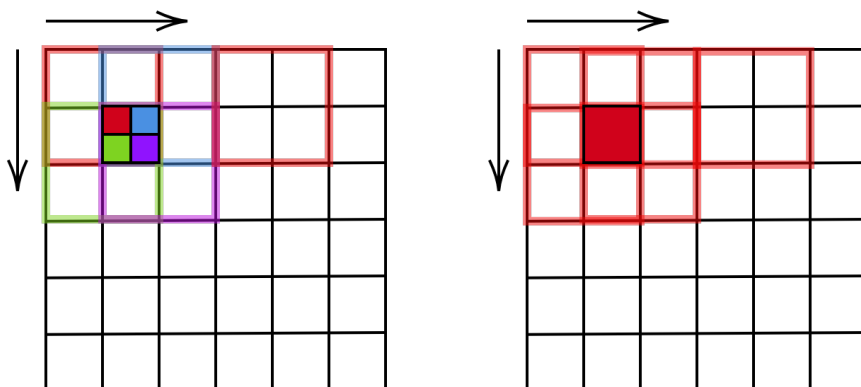


3.4. att. *InternImage* bloka shēma (veidots balstoties uz (W. Wang, J. Dai et al., 2022))

3.3.3 Jaunieviestais mehānisms *LightDCN*

LightDCN ir mūsu izveidots mehānisms, kas modificē DCN, un tā pamata ideja ir samazināt DCN apmācāmo parametru skaitu, liekot blakus esošām kodolu pozīcijām izmantot tās pašas nobīdes. Attēlā 3.5. redzams salīdzinājums starp kodolu nobīžu daudzumu dažādās pozīcijās. Ja ir ieejas attēls vai iezīmju karte, tad DCN mehānisms iet pāri tam konvolūcijas manierē, un nobīda kodola šūnas balstoties uz nobīdēm attiecīgajās vietās. Kā redzams kreisajā pusē uz iekrāsotā pikseļa pārklājas 4 dažādi kodoli, jo to izmērs ir 2×2 . Tātad attiecīgajā attēla pozīcijā ir 4 dažādes x un y ass nobīdes, katram atsevišķajam kodolam. Savukārt labajā pusē *LightDCN* mehānismā šai pozīcijai ir tikai viena x un y nobīdes vērtība, un kodoli, kas pārklājas uz šīs pozīcijas, ņem attiecīgās nobīdes vērtības. Formulā 1.14. ir matemātiski aprakstīta deformējamo konvolūciju mehānisms, un nobīžu matrica ir $\Delta P \in \mathbb{R}^{2N \times H \times W}$, un tās kanālu skaits ir $2N$. Attēlotajā gadījumā $N = 4$, jo kodolam ir $2 \times 2 = 4$ šūnas. *LightDCN* nobīžu matrica ir $\Delta P \in \mathbb{R}^{2 \times H \times W}$, un to var aprakstīt šādi:

$$\begin{aligned} \Delta p(l) &= \Delta P_{l_y, l_x} \\ y(p) &= \sum_{p_n \in \mathcal{O}} w(p_n) \cdot x(p + p_n + \Delta p(p + p_n)) \end{aligned} \quad (3.8)$$



3.5. att. *LightDCN* mehānisma shēma. Kreisajā pusē DCN(J. Dai, H. Qi et al., 2017) kodolu nobīžu daudzuma ilustrācija, labajā *LightDCN*

3.4. Apmācības un testēšanas protokols

Originālā pētījuma rezultātu atkārtošanai un izveidoto arhitektūru apmācībai, tika izmantotas *mmpretrain*(Contributors, 2023) un *mmdetection*(K. Chen, J. Wang et al., 2019) pakotnes. Tīklu implementācijas ir rakstītas izmantojot *PyTorch*(Paszke, Gross et al., 2017) pakotni, un izmanto iepriekšējo pētījumu izveidotas bloku implementācijas. Eksperimentu un izveidotu modeļu pirmkods ir pieejams <https://github.com/march-o/deform-conv>.

Eksperimenti tiek veikti uz *Small-ImageNet*(3.1. nodaļa) datu kopas. Aplūkotie modeļi tiek apmācīti 100 epochas(angļu val. *epoch*), un tiek veikta novērtēšana uz validācijas kopas pēc katra epoha. Apmācības laikā pirms attēla padošanas modelim, tiek veikta sekojošā priekšapstrāde:

1. Attēla apgriešana pēc nejaušības principa, un izmēra maiņa uz 224×224 pikseļiem,
2. Ar varbūtību 0.5 attēls tiek apgriezts horizontāli,
3. Attēla krāsu vērtības tiek normalizētas(tiek izmantotas aprēķinātās krāsu vidējās vērtības un standartnovirzes vērtības *Small-ImageNet* datu kopā),
4. Krāsu vērtības tiek standartizētas,
5. Izmantojot partijas izmēru 256, attēli tiek grupēti.

Apmācībai tiek izmantota viena *NVIDIA RTX A6000* GPU, izņemot rezultātu atkārtošanai tika izmantotas astoņas *NVIDIA L4* GPU, lai apmācības vide būtu tāda pati kā oriģinālajā pētījumā. Senākām un vienkāršākām arhitektūrām kā *ResNet* eksperimentu ilgums bija apmēram 4 stundas, tomēr sarežģītākām

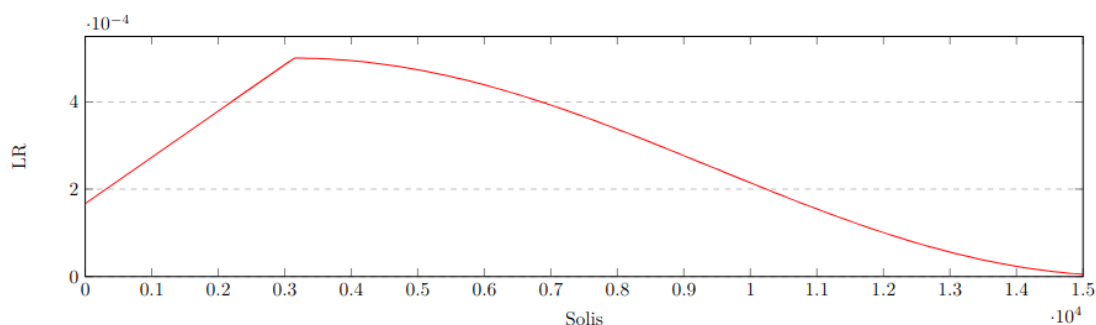
arhitektūrām, kur skaitlisko operāciju paralelizācija nav tik vienkārša vai pat iespējama, prasīja virs 7 stundām.

3.4.1 Izmantotie optimizētāji un mācīšanās ātruma plānotāji

Balstoties uz oriģinālo *ResNet* implementāciju (He, X. Zhang et al., 2015), 3.3.1. nodaļā aprakstītie modeļu parametri tika optimizēti izmantojot *Adam* (Kingma & Ba, 2014) algoritmu. Tiek izmantots statisks LR, un katrai arhitektūrai tie tika noteikti izmantojot metodi aprakstītu 3.4.2. nodaļā.

3.3.2. un 3.3.3. nodaļās aprakstītie modeļi izmanto modernākus un sarežģītākus slāņus, tādēļ to apmācībai ir jāizmanto citi algoritmi. Ņemot piemēru no (W. Wang, J. Dai et al., 2022) tiek izmantots *AdamW* (Loshchilov & Hutter, 2017) optimizācijas algoritms un LR plānotājs ar sekojošajiem posmiem (3.6. attēls):

- No 0 līdz 20 epocham tiek izmantots lineārs mācīšanās ātruma palielinājums no $1.7 \cdot 10^{-4}$ līdz $5 \cdot 10^{-4}$,
- no 20 līdz 100 epocham tiek izmantota kosinusa LR samazināšana (Loshchilov & Hutter, 2016) ar beigu LR $1 \cdot 10^{-6}$.



3.6. att. LR vērtība pret apmācības soli

3.4.2 Mācīšanās ātruma atrašana

Mācīšanās ātrums (angļu val. *learning rate*) ir svarīgākais hiperparametrs (angļu val. *hyperparameter*) tīkla apmācības procesam. Tas nosaka cik strauji tiek mainīti modeļa svāri, algoritmā 1.1. apzīmēts ar η . Pārāk zems un tīkla svāri netiek pietiekami *stipri* modificēti, lai to vērtības atrastos zuduma funkcijas minimumā. Pārāk augsts mācīšanās ātrums var novest pie nestabila svaru modifikācijas procesa un tas var nebeidzami pāršaut optimālo minimumu, mainot svaru vērtības nepareizajā virzienā. (Smith, 2015) piedāvā metodi kā atrast optimālu mācīšanās ātrumu. Katru apmācības soli eksponenciāli tiek palielināts mācīšanās ātrums un aprēķināta zuduma funkcijas vērtība. Vietā, kur visātrāk kritās

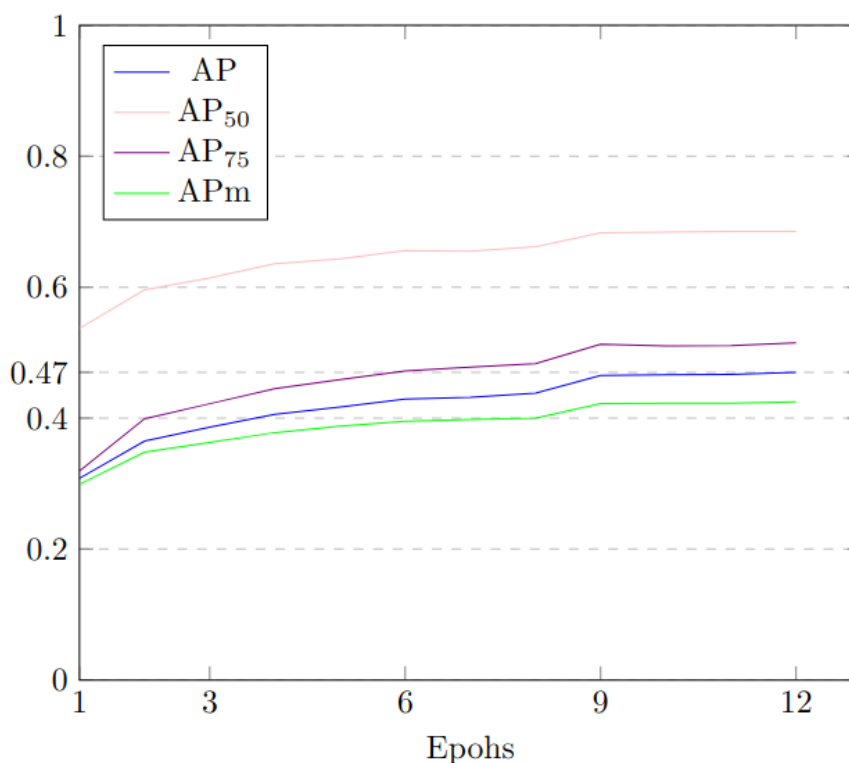
zuduma funkcijas vērtības jeb tai ir vismazākā gradienta vērtība, ir apmēram atbilstoša mācīšanās ātruma vērtība. Mēs izmantojam šo metodi daļā no saviem eksperimentiem, lai nebūtu jāveic ilgas hiperparametru pārmeklēšanas un iegūtu dziļāku ieskatu tīkla konverģences stabilitātē.

4. REZULTĀTI

Šajā nodaļā aplūkoti bakalaura darba praktiskās daļas ietvaros iegūtie rezultāti. Sākumā ir aprakstīta rezultātu atkārtošana, turpinot veidoto arhitektūru apmācības eksperimenti, kā arī dziļāks ieskats deformējamo konvolūciju mehānisma darbībā attiecīgos piemēros.

4.1. Deformējamo konvolūciju modeļa apmācības atkārtošana

Vadoties pēc bakalaura darba uzdevumiem, pirmais no eksperimentiem bija rezultātu atkārtošana objektu atpazīšanas uzdevumā ar nemainītu deformējamo konvolūcijas modeli. Modeļa kods tika ņemts no (W. Wang, J. Dai et al., 2022) atvērtā pirmkoda. Modelis tika apmācīts 12 epohas uz MS COCO datu kopas. Visi apmācības parametri tika atstāti tādi paši kā pētījumā. Aplūkojot rādītājus 4.1. attēlā, var secināt, ka rezultātus ir izdevies atkārtot, iegūstot tādu pašu AP kā pētījumā 0.47.



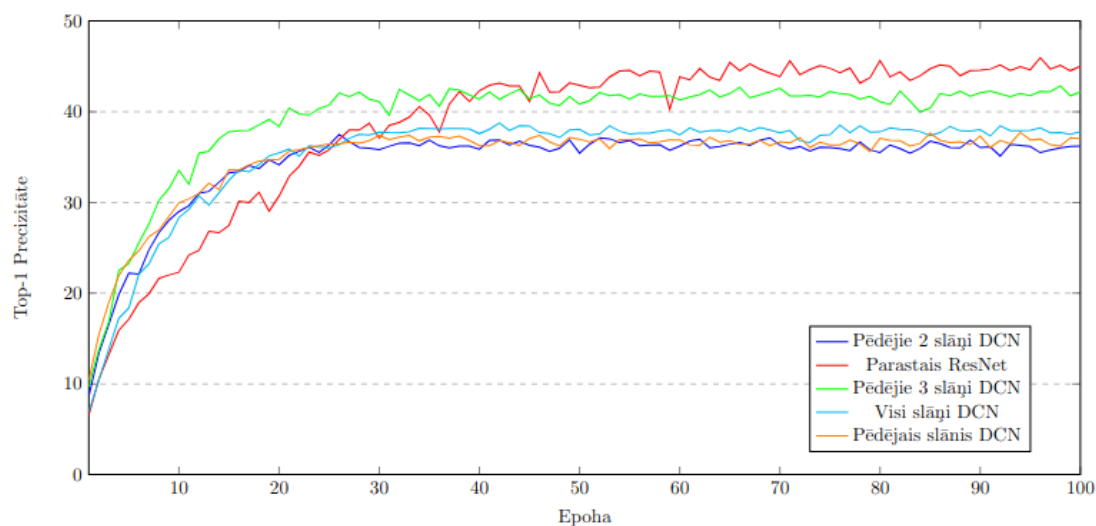
4.1. att. Objektu atpazīšanas rādītāji modeļa treniņa laikā

4.2. Eksperimenti attēlu klasifikācijas uzdevumā

Šajā nodaļā tika aplūkoti iegūtie rezultāti no eksperimentiem, kas aprakstīti 3.3. nodaļā attēlu klasifikācijas uzdevumā.

4.2.1 *ResNet* ar deformējamajām konvolūcijām

Tā kā tika izmantota jauna apmācības kopa un izveidotas jaunas arhitektūras, lai atrastu piemērotu LR, tika izmantota metode, kas aprakstīta 3.4.2. nodaļā. Pielikumā var aplūkot meklēšanas zuduma funkcijas līknes, izvēlētie LR ir pierakstīti 4.1. tabulā. Attēlā 4.3. redzami atšķirīgo arhitektūru apmācības procesa rādītāji. Kā var redzēt modeļi ar DCN konverģē ievērojami ātrāk, kas varētu būt saistīts ar to spēju iemācīties reģionus, kur *skatīties* ātrāk. Tomēr aplūkojot 4.1. tabulu var novērot, ka parastais *ResNet* modelis iegūst vislabākos rezultātus. Tas varētu būt skaidrojams ar DCN modeļu pārmērīgu pielāgošanos apmācības datu kopai un nespēju vispārināties. Starp modificētajiem modeļiem, variants ar 3 pēdējiem DCN slāņiem iegūst vislabākos rezultātus gan rādītājos, gan konverģences ātrumā. Šis rezultāts norāda uz to, ka parastā konvolūcija ir piemērota sākuma attēla apstrādei, atrodot prastākās iezīmes. Deformējamās konvolūcijas, specifiski to atkārtota izmantošana, labi spēj atrast sarežģītākas attēla iezīmes, tas ir tādas, kur nepieciešams lielāks attēla konteksts.



4.2. att. *ResNet* ar DCN modeļu apmācības rādītāji

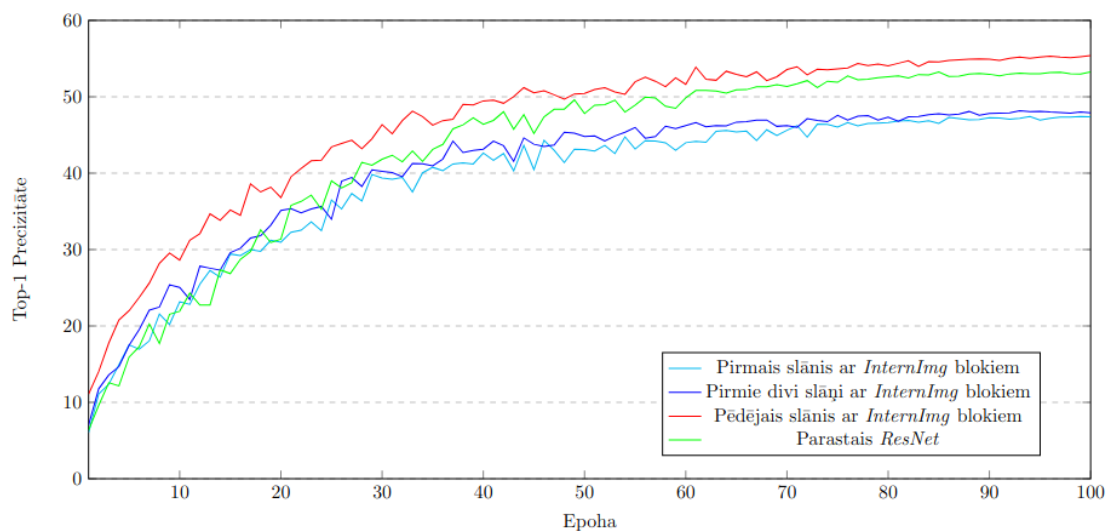
4.1. tabula

***ResNet* modeļa ar DCN izmantošanu apmācības rādītāji**

DCN pēdējo slāņu skaits	Top-1 prec. %	Top-5 prec. %	LR	GPU atmiņas patēriņš (MB)
0	45.95	71.49	$2 * 10^{-4}$	21496
1	37.74	63.94	$7 * 10^{-5}$	21506
2	37.51	63.71	$4 * 10^{-5}$	21563
3	42.84	68.25	$7 * 10^{-5}$	21712
4	38.75	64.75	$5 * 10^{-5}$	22117

4.2.2 *ResNet* ar *InternImage* blokiem

Attēlā 4.3. redzami 3.3.2. nodaļā aprakstīto arhitektūru apmācības rezultāti. Var novērot, ka izmantojot modernāku apmācības procesu, kas aprakstīts 3.4. nodaļā, modeļu konverģence ir lēnākā, bet tie sasniedz augstākus rezultātus. Tieši salīdzinot parastos *ResNet* tīklus, var redzēt, ka konverģences ātrums ir nedaudz lēnāks, bet tiek uzstādīti labāki rezultāti. Tabulā 4.2. redzams, ka arhitektūra ar pēdējo slāni no *InternImage* blokiem iegūst visaugstākos rezultātus. Tas parāda, ka kombinējot DCN ar modernākām CNN arhitektūrām un apmācības metodoloģijām ir iespējams iegūt labākus rezultātus. Tomēr jāpiebilst, ka iepriekšminētā modeļa GPU atmiņas patēriņš ir ievērojami lielāks.



4.3. att. *ResNet* ar *InternImage* blokiem modeļu apmācības rādītāji

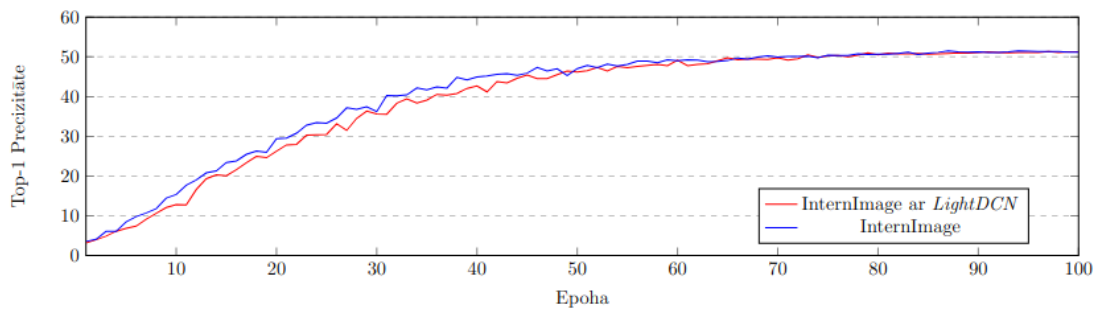
4.2. tabula

ResNet modeļa ar *InternImage* bloku izmantošanu apmācības rādītāji

<i>InternImg</i> slāņu konfigurācija	Top-1 prec. %	Top-5 prec. %	GPU atmiņas patēriņš (MB)
0-0-0-0	53.23	76.91	21473
1-0-0-0	47.41	71.65	13631
1-1-0-0	48.16	71.82	21923
0-0-0-1	55.34	77.57	28194

4.2.3 *LightDCN*

Attēlā 4.4. redzami apmācības procesa rādītāji, te var redzēt, ka tiek iegūti apmēram vienādi rezultāti. Tabulā 4.3. var redzēt, ka *LightDCN* parametru skaits ir par 2 miljoniem(6%) mazāks. Tātad šis mehānisms uzstāda tāds pašus rezultātus izmantojot ievērojami mazāk resursus.



4.4. att. *InternImage* ar DCN un *LightDCN* mehānismu apmācības rādītāji

4.3. tabula

DCN un *LightDCN* modeļu izmēra salīdzinājums. Rādītāji tika iegūti izmantojot *mmpretrain*(Contributors, 2023) pakotni

Mehānisms	Peldošā punkta operāciju skaits (miljardos)	Parametru skaits (miljonos)	Parametru skaits %
<i>DCN</i>	4.903	29.303	100 %
<i>LightDCN</i>	4.518	27.601	94 %

4.3. Eksperimenti objektu atpazīšanas uzdevumā

Sekojošajās metodēs, kas tika aplūkotas literatūras analīzē, pēc modeļu apmācības attēlu klasifikācijas uzdevumā tā svāri tika izmantoti objektu atpazīšanas uzdevumā. Lielākā daļa no modeļu parametriem, izņemot pēdējo lineāro slāni, kas nosaka attēla klasi, tiek izmantoti, lai iegūtu galvenās attēla iezīmes. Tika izmantota nozarē un citos pētījumos bieži izmantota objektu atpazīšanas galva - *MaskRCNN* (He, Gkioxari et al., 2017). Viss modelis tiek apmācīts uz *coco-minitrain* (Samet, Hicsonmez et al., 2020) datu kopas izmantojot *mmdetection* (K. Chen, J. Wang et al., 2019) pakotni.

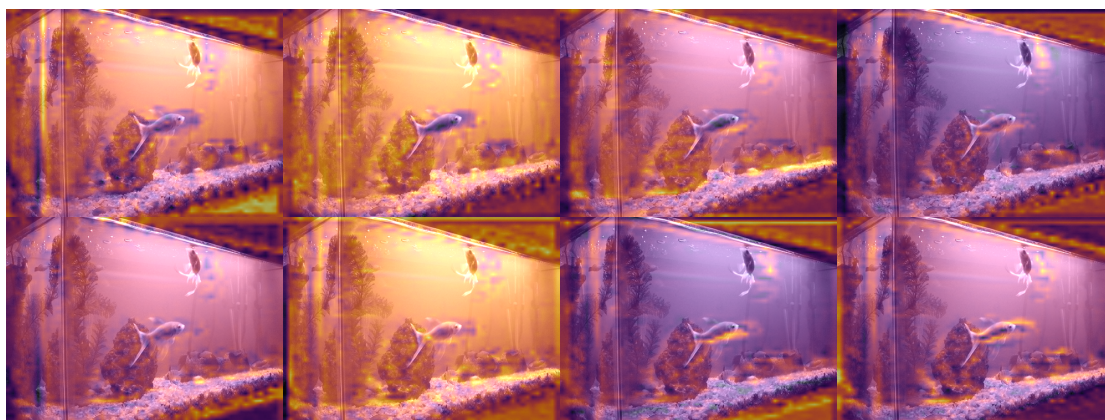
Veicot eksperimentus, kas apmāca 3.3.2. nodaļā minētos modeļus, tika novērots, ka modelis, kurš attēlu klasifikācijas uzdevumā uzstādīja augstākus veikspējas rādītājus, objektu atpazīšanu veic ievērojami sliktāk. Tika iegūts uz pusi mazāks AP (3.2. nodaļa) par parasto *ResNet* arhitektūru. Šāds veikspējas kritums varētu būt izskaidrojams ar nepareizu svaru pārneses vai apmācības implementāciju. Ir arī iespējams, ka iemesls ir daudz sarežģītāks un šāda arhitektūra nav piemērota objektu atpazīšanas uzdevumam. Atšķirībā no attēlu klasifikācijas, kur visa informācija, ko izmanto modeļa galva, iet cauri visam iezīmju izgūvējam, objektu atpazīšanas uzdevumā modeļa galva izmanto informāciju, ko atgriež katrs no slāņiem. Ņemot šo vērā, ja kāds no slāņiem būtiski atšķiras no pārējiem (kā tas ir aplūkotajās arhitektūrās), tad ir iespējams, ka šis ļoti ievērojami samazina modeļa galvas veikspēju, jo tā nav veidota šādiem nolūkiem.

4.4. Kodolu deformējāciju apskats un salīdzinājums

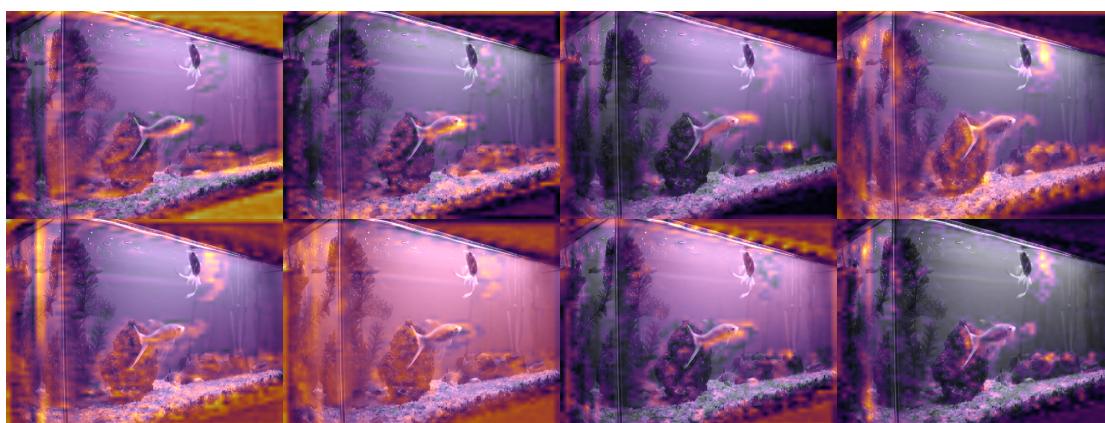
Šajā nodaļā tiks dziļāk aplūkotas DCN un *LightDCN* kodolu šūnu nobīdes pēc modeļu apmācības. Apmācības sākumā atšķirībā no citiem parametriem, kuri tiek inicializēti ar nejaušām vērtībām, deformācijas parametri tiek inicializēti ar nullēm, lai sākumā to darbība būtu tāda pati kā parastajām konvolūcijām. Attēlā 4.5. redzamas katra pikseļa pārklājošo kodolu nobīžu absolūto vērtību summa atsevišķi pa x un y asīm.

Līdzīgi attēlā 4.6. redzami jaunievietā mehānisma *LightDCN* nobīdes. Kā minēts 3.3.3. nodaļā, šis mehānisms liek kodoliem, kas pārklājas, izmantot tās pašas nobīdes. Tādēļ ir attēlotas katra pikseļa nobīžu absolūtās vērtības atsevišķi pa x un y asīm.

Var novērot, ka *LightDCN* mehānisms mazāk veic deformācijas kopā un arī reģionos, kur neatrodas svarīgi objekti. Tomēr nozīmīgajās vietās abi mehānismi veic deformācijas, lai iegūtu vairāk informācijas par attiecīgo reģionu.



4.5. att. DCN kodolu šūnu nobīdes. Augšā nobīdes pa x asi, apakšā pa y. Horizontāli dažādas nobīžu grupas (ILSVRC2012 (Deng, Dong et al., 2009) validācijas kopas piemērs nr. 00000262)



4.6. att. *LightDCN* kodolu šūnu nobīdes. Augšā nobīdes pa x asi, apakšā pa y. Horizontāli dažādas nobīžu grupas (ILSVRC2012 (Deng, Dong et al., 2009) validācijas kopas piemērs nr. 00000262)

SECINĀJUMI

Darba ietvaros tika aplūkots kā deformējamo konvolūciju mehānisms ietekmē dziļajā mašīnmācīšanās balstītu attēlu klasifikācijas un objektu atpazīšanas sistēmu veikspēju. Kā arī tiek piedāvāts jauns mehānisms *LightDCN*, kas ir deformējamo konvolūciju modifikācija, un tiek salīdzināts ar parasto mehānismu.

Tika veikta sistemātiska literatūras analīze, lai atrastu pētījumus, kur tiek piedāvātas jaunas metodes attēlu klasifikācijai un citu datorredzes uzdevumu izpildei. Pēc analīzes kvantitatīvā modeļu salīdzinājumā tiek atrasts, ka arhitektūras, kas uzstāda augstākos veikspējas rādītājus, ir *VOLO* (Yuan, Hou et al., 2021) un *CAFormer* (Yu, Si et al., 2022).

Nemot vērā pieejamo skaitļošanas budžetu tika izveidota jauna datu kopā *Small-ImageNet*, kas ir apakškopa *ImageNet* (Deng, Dong et al., 2009) datu kopai. Šī datu kopa ļāva veikt vairāk un ātrākus ablācijas pētījumus un citus eksperimentus. Tās izveides kods ir pieejams šī darba atvērtajā pirmkodā.

Salīdzinot deformējamo konvolūciju mehānismu attēlu klasifikācijas uzdevumā ar *Small-ImageNet* datu kopu, tika atrasts, ka pievienojot DCN, lai arī konverģences ātrums ievērojami palielinās, veikspēja tikai pasliktinās. Parastais *ResNet* modelis iegūst visaugstākos rezultātus - 45.95% precizitāti. Otrais augstākais ir variants, kur pēdējie 3 no 4 slāņiem izmanto deformējamās konvolūcijas un iegūst 42.84% precizitāti, apliecinot, ka izmantojot parastās konvolūcijas attēla sākuma apstrādei, palīdz DCN.

Tika aplūkota arī modernu mašīnmācīšanās slāņu un apmācības procesa kombinēšana ar DCN un *ResNet*. Šajos eksperimentos visaugstākos rezultātus uzrādīja modelis, kas izmanto deformējamās konvolūcijas pēdējā slānī, iegūstot 55.34% precizitāti.

Objektu atpazīšanas uzdevumā aplūkotās arhitektūras ieguva ievērojami sliktākus rezultātus kā parastā *ResNet* arhitektūra. Viens no iemesliem, kas varētu traucēt modeļu veikspējai, ir to nehomogēnā arhitektūra, kas izmanto atšķirīgus mehānismus.

LightDCN ir šī darba ietvaros izstrādāts jauns mehānisms, kas seko konvolūcijas deformēšanas idejai, bet modificē daļu no mehānisma darbības. Apmācot parasto DCN un *LightDCN* veikspējas rādītāji ir vienādi, bet modeļa parametru skaits sarūk no 29.303 miljoniem uz 27.601 (94%). Šis mehānisms ir vieglāks DCN variants un to ir vērts pētīt dziļāk. Kā arī tiek aplūkota abu mehānismu darbība atsevišķos piemēros, un var novērot, ka *LightDCN* veic mazāk deformācijas, bet saglabā to darbību nozīmīgos reģionos.

IZMANTOTIE INFORMĀCIJAS AVOTI

- Ba, Jimmy, Jamie Ryan Kiros et al., “Layer Normalization”. *ArXiv* abs/1607.06450 (2016). Pieejams: <https://api.semanticscholar.org/CorpusID:8236317>.
- Bahdanau, Dzmitry, Kyunghyun Cho et al., “Neural Machine Translation by Jointly Learning to Align and Translate”. *CoRR* abs/1409.0473 (2014). Pieejams: <https://api.semanticscholar.org/CorpusID:11212020>.
- Bishop, Christopher M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1. izdev. Springer, 2007. ISBN: 0387310738.
- Cai, Yuxuan, Yi Zhou et al., “Reversible Column Networks”. *ArXiv* abs/2212.11696 (2022).
- Cai, Zhaowei & Nuno Vasconcelos. “Cascade R-CNN: High Quality Object Detection and Instance Segmentation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43 (2019), 1483.—1498. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:195345409>.
- Carion, Nicolas, Francisco Massa et al., “End-to-End Object Detection with Transformers”. *ArXiv* abs/2005.12872 (2020).
- Caron, Mathilde, Hugo Touvron et al., “Emerging Properties in Self-Supervised Vision Transformers”. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 9630.—9640. lpp.
- Chen, Kai, Jiaqi Wang et al., *MMDetection: Open MMLab Detection Toolbox and Benchmark*. 2019. arXiv: 1906.07155 [cs.CV].
- Chen, Yinpeng, Xiyang Dai et al., “Dynamic Convolution: Attention Over Convolution Kernels”. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 11027.—11036. lpp.
- Cheng, Bowen, Ishan Misra et al., “Masked-attention Mask Transformer for Universal Image Segmentation”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 1280.—1289. lpp.
- Chollet, François. “Xception: Deep Learning with Depthwise Separable Convolutions”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), 1800.—1807. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:2375110>.
- Contributors, MPreTrain. *OpenMMLab’s Pre-training Toolbox and Benchmark*. <https://github.com/open-mmlab/mmpretrain>. 2023.
- contributors, PapersWithCode. *Papers with Code: A Community and Resource for Machine Learning Research*. <https://github.com/paperswithcode>. 2017.

- Dai, Jifeng, Haozhi Qi et al., “Deformable Convolutional Networks”. *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), 764.—773. lpp.
- Dai, Zihang, Hanxiao Liu et al., “CoAtNet: Marrying Convolution and Attention for All Data Sizes”. *ArXiv* abs/2106.04803 (2021).
- Deng, Jia, Wei Dong et al., “Imagenet: A large-scale hierarchical image database”. *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, 248.—255. lpp.
- Ding, Mingyu, Bin Xiao et al., “DaViT: Dual Attention Vision Transformers”. (2022), 74.—92. lpp.
- Ding, Xiaohan, Yiyuan Zhang et al., “UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition”. *ArXiv* abs/2311.15599 (2023).
- Dosovitskiy, Alexey, Lucas Beyer et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. *ArXiv* abs/2010.11929 (2020).
- Everingham, Mark, Luc Van Gool et al., “The Pascal Visual Object Classes (VOC) Challenge.” *Int. J. Comput. Vis.* 88.2 (2010), 303.—338. lpp. Pieejams: <http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWZ10>.
- Fellbaum, Christiane. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998. Pieejams: <https://mitpress.mit.edu/9780262561167/>.
- Fukushima, Kunihiko. “Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”. *Biological Cybernetics* 36 (1980), 193.—202. lpp.
- “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements”. *IEEE Trans. Syst. Sci. Cybern.* 5 (1969), 322.—333. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:206799280>.
- Goodfellow, Ian J., Yoshua Bengio et al., *Deep Learning*. <http://www.deeplearningbook.org>. Cambridge, MA, USA: MIT Press, 2016.
- Hassani, Ali & Humphrey Shi. “Dilated Neighborhood Attention Transformer”. *ArXiv* abs/2209.15001 (2022).
- Hatamizadeh, Ali, Hongxu Yin et al., “Global Context Vision Transformers”. (2022), 12633.—12646. lpp.
- He, Kaiming, Georgia Gkioxari et al., “Mask R-CNN”. 2017. Pieejams: <https://api.semanticscholar.org/CorpusID:54465873>.
- He, Kaiming, X. Zhang et al., “Deep Residual Learning for Image Recognition”. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

- (2015), 770.—778. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:206594692>.
- Hendrycks, Dan & Kevin Gimpel. “Gaussian Error Linear Units (GELUs)”. *arXiv: Learning* (2016). Pieejams: <https://api.semanticscholar.org/CorpusID:125617073>.
- Hubel, David H. & Torsten N. Wiesel. “Receptive Fields of Single Neurons in the Cat’s Striate Cortex”. *Journal of Physiology* 148 (1959), 574.—591. lpp.
- Ioffe, Sergey & Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. *ArXiv* abs/1502.03167 (2015). Pieejams: <https://api.semanticscholar.org/CorpusID:5808102>.
- Kim, Jinpyo, Wookeun Jung et al., “CyCNN: A Rotation Invariant CNN using Polar Mapping and Cylindrical Convolution Layers”. *ArXiv* abs/2007.10588 (2020).
- Kingma, Diederik P. & Jimmy Ba. “Adam: A Method for Stochastic Optimization”. *CoRR* abs/1412.6980 (2014). Pieejams: <https://api.semanticscholar.org/CorpusID:6628106>.
- Kinney, Rodney Michael, Chloe Anastasiades et al., “The Semantic Scholar Open Data Platform”. *ArXiv* abs/2301.10140 (2023). Pieejams: <https://api.semanticscholar.org/CorpusID:256194545>.
- Krizhevsky, Alex, Ilya Sutskever et al., “ImageNet Classification with Deep Convolutional Neural Networks”. *Advances in Neural Information Processing Systems* 25. Izdevis F. Pereira, C. J. C. Burges et al., Curran Associates, Inc., 2012, 1097.—1105. lpp. Pieejams: <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- LeCun, Y., B. Boser et al., “Backpropagation Applied to Handwritten Zip Code Recognition”. *Neural Computation* 1 (1989), 541.—551. lpp.
- LeCun, Yann & Corinna Cortes. “MNIST handwritten digit database”. (2010). Pieejams: <http://yann.lecun.com/exdb/mnist/>.
- Li, Yanghao, Chaoxia Wu et al., “MViTv2: Improved Multiscale Vision Transformers for Classification and Detection”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 4794.—4804. lpp.
- Lin, Tsung-Yi, Michael Maire et al., “Microsoft COCO: Common Objects in Context”. *European Conference on Computer Vision*. 2014. Pieejams: <https://api.semanticscholar.org/CorpusID:14113767>.
- Lin, Wei-Shiang, Ziheng Wu et al., “Scale-Aware Modulation Meet Transformer”. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 5992.—6003. lpp.

- Liu, Jihao, Hongsheng Li et al., “UniNet: Unified Architecture Search with Convolution, Transformer, and MLP”. (2021), 33.—49. lpp.
- Liu, Shiwei, Tianlong Chen et al., “More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity”. *ArXiv* abs/2207.03620 (2022).
- Liu, Ze, Yutong Lin et al., “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 9992.—10002. lpp.
- Liu, Zhuang, Hanzi Mao et al., “A ConvNet for the 2020s”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 11966.—11976. lpp.
- Loshchilov, Ilya & Frank Hutter. “Decoupled Weight Decay Regularization”. *International Conference on Learning Representations*. 2017. Pieejams: <https://api.semanticscholar.org/CorpusID:53592270>.
- “SGDR: Stochastic Gradient Descent with Warm Restarts”. *arXiv: Learning* (2016). Pieejams: <https://api.semanticscholar.org/CorpusID:14337532>.
- Luong, Thang, Hieu Pham et al., “Effective Approaches to Attention-based Neural Machine Translation”. *ArXiv* abs/1508.04025 (2015). Pieejams: <https://api.semanticscholar.org/CorpusID:1998416>.
- mnmostafa, Mohammed Ali. *Tiny ImageNet*. 2017. Pieejams: <https://kaggle.com/competitions/tiny-imagenet>.
- Mo, Hanlin & Guoying Zhao. “RIC-CNN: Rotation-Invariant Coordinate Convolutional Neural Network”. *Pattern Recognit.* 146 (2022), 109994. lpp.
- El-Nouby, Alaaeldin, Hugo Touvron et al., “XCiT: Cross-Covariance Image Transformers”. (2021), 20014.—20027. lpp.
- Parikh, Ankur P., Oscar Täckström et al., “A Decomposable Attention Model for Natural Language Inference”. *ArXiv* abs/1606.01933 (2016). Pieejams: <https://api.semanticscholar.org/CorpusID:8495258>.
- Paszke, Adam, Sam Gross et al., “Automatic differentiation in PyTorch”. (2017).
- Qi, Qi, Yan Yan et al., “A Simple and Effective Framework for Pairwise Deep Metric Learning”. *Computer Vision – ECCV 2020*. Izdevis Andrea Vedaldi, Horst Bischof et al., Cham: Springer International Publishing, 2020, 375.—391. lpp.
- Rao, Yongming, Wenliang Zhao et al., “HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions”. *ArXiv* abs/2207.14284 (2022).

- Roberts, Lawrence G. *Machine Perception of Three-Dimensional Solids*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 2002. g. 24. maijs. ISBN: 0-8240-4427-4.
- Rosenblatt, Frank. “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review* 65 6 (1958), 386.—408. lpp.
- Rumelhart, David E, Geoffrey E Hinton et al., “Learning representations by back-propagating errors”. *nature* 323.6088 (1986), 533.—536. lpp.
- Samet, Nermin, Samet Hicsonmez et al., “HoughNet: Integrating near and long-range evidence for bottom-up object detection”. *European Conference on Computer Vision (ECCV)*. 2020.
- Schmidhuber, Jürgen. “Learning to Control Fast-Weight Memories: An Alternative to Dynamic Recurrent Networks”. *Neural Computation* 4 (1992), 131.—139. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:16683347>.
- Shi, Dai. “TransNeXt: Robust Foveal Visual Perception for Vision Transformers”. *ArXiv abs/2311.17132* (2023).
- Simonyan, Karen & Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. *CoRR abs/1409.1556* (2014). Pieejams: <https://api.semanticscholar.org/CorpusID:14124313>.
- Smith, Leslie N. “Cyclical Learning Rates for Training Neural Networks”. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2015), 464.—472. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:15247298>.
- Tan, Mingxing & Quoc V. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. *ArXiv abs/1905.11946* (2019).
- Touvron, Hugo, M. Cord et al., “Going deeper with Image Transformers”. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 32.—42. lpp.
- Tu, Zhengzhong, Hossein Talebi et al., “MaxViT: Multi-Axis Vision Transformer”. (2022), 459.—479. lpp.
- Vaillant, R., C. Monrocq et al., “Original Approach for the Localisation of Objects in Images”. *IEE Proceedings - Vision, Image and Signal Processing* 141.4 (1994), 245.—250. lpp.
- Vaswani, Ashish, Prajit Ramachandran et al., “Scaling Local Self-Attention for Parameter Efficient Visual Backbones”. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 12889.—12899. lpp.

- Vaswani, Ashish, Noam M. Shazeer et al., “Attention is All you Need”. *Neural Information Processing Systems*. 2017. Pieejams: <https://api.semanticscholar.org/CorpusID:13756489>.
- Wang, Peng, Shijie Wang et al., “ONE-PEACE: Exploring One General Representation Model Toward Unlimited Modalities”. *ArXiv* abs/2305.11172 (2023).
- Wang, Wenhai, Jifeng Dai et al., “InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions”. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 14408.—14419. lpp.
- Wightman, Ross & PyTorch Image Models contributors. *PyTorch Image Models*. <https://github.com/rwightman/pytorch-image-models>. 2019. Pieejams: doi: 10.5281/zenodo.4414861.
- Xia, Zhuofan, Xuran Pan, S. Song et al., “Vision Transformer with Deformable Attention”. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 4784.—4793. lpp.
- Xia, Zhuofan, Xuran Pan, Shiji Song et al., “DAT++: Spatially Dynamic Vision Transformer with Deformable Attention”. *ArXiv* abs/2309.01430 (2023).
- Xiao, Tete, Yingcheng Liu et al., “Unified Perceptual Parsing for Scene Understanding”. *ArXiv* abs/1807.10221 (2018). Pieejams: <https://api.semanticscholar.org/CorpusID:50781105>.
- Xiong, Yuwen, Zhiqi Li et al., “Efficient Deformable ConvNets: Rethinking Dynamic and Sparse Operator for Vision Applications”. *ArXiv* abs/2401.06197 (2024).
- Yang, Jianwei, Chunyuan Li et al., “Focal Modulation Networks”. *ArXiv* abs/2203.11926 (2022).
- Yu, Weihao, Chenyang Si et al., “MetaFormer Baselines for Vision”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46 (2022), 896.—912. lpp.
- Yuan, Li, Qibin Hou et al., “VOLO: Vision Outlooker for Visual Recognition”. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2021), 6575.—6586. lpp.
- Zhang, Hao, Feng Li et al., “DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection”. *ArXiv* abs/2203.03605 (2022).
- Zhou, Bolei, Hang Zhao et al., “Scene Parsing through ADE20K Dataset”. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), 5122.—5130. lpp. Pieejams: <https://api.semanticscholar.org/CorpusID:5636055>.
- Zhou, Daquan, Yujun Shi et al., “Refiner: Refining Self-attention for Vision Transformers”. *ArXiv* abs/2106.03714 (2021).

- Zhu, Xizhou, Han Hu et al., “Deformable ConvNets V2: More Deformable, Better Results”. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 9300.—9308. lpp.
- Zhu, Xizhou, Weijie Su et al., “Deformable DETR: Deformable Transformers for End-to-End Object Detection”. *ArXiv* abs/2010.04159 (2020).
- Zong, Zhuofan, Guanglu Song et al., “DETRs with Collaborative Hybrid Assignments Training”. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (2022), 6725.—6735. lpp.

PIELIKUMI

ResNet ar DCN modeļu LR hiperparametra meklēšanas rezultāti

