

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**CEĻA SEGUMA IEZĪMJU NOTEIKŠANA, IZMANTOJOT  
MAŠĪNMĀCĪŠANOS**

BAKALaura DARBS

Autors: **Krišjānis Šules**  
Studenta apliecības Nr.: ks10163  
Darba vadītājs: M. Dat. Muntis Rudzītis

RĪGA 2018

## ANOTĀCIJA

Pēdējos gados pastiprināta uzmanība tiek pievērsta datorredzes problēmām un konvolūciju neironu tīklu lomai problēmu risinājumos. Darba teorētiskajā daļā vispārīgi tiek apskatīta mākslīgo neironu uzbūve un vēsture, kā arī konvolūciju neironu tīklu uzbūve un jaunākie konvolūciju neironu tīklu modeļi, t.s. “reģionu” bāzētie: R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN un “viena šāviena detektora” bāzētie: SSD un YOLO. Darbā tiek pētītas ceļu seguma iezīmes un to izmantošana neironu tīklu apmācīšanā, kā arī tiek sniegts neliels ieskats mašīnmācīšanās ietvaru izvēlē. Tiek apskatīti arī citi pētījumi, kas ir saistīti ar šo tēmu un alternatīvu risinājumu izveidi. Darba nobeigumā tiek aprakstīti secinājumi par konvolūciju neironu tīklu veidu izvēli atbilstoši ceļa seguma iezīmju noteikšanai, kā arī citi apsvērumi praktiska risinājuma izstrādei.

Atslēgvārdi: neironu tīkli, konvolūciju neironu tīkli, objektu noteikšana, ceļa seguma iezīmju noteikšana, datorredze, dziļās mašīnmācīšanās ietvari.

## **ABSTRACT**

### **ROAD SURFACE FEATURE DETECTION WITH MACHINE LEARNING**

In recent years attention has increased for computer vision problems and the role of convolutional neural networks in solutions. The theoretical part of this study includes a general overview of artificial neural networks, convolutional neural networks and latest convolutional neural network models such as, region-based: R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN and one shot detector based: SSD and YOLO. The work also contains study of road surface features and their use in neural network training. A brief overview is provided about a number of machine learning frameworks with recommendations. Several other papers related to this topic and alternative solutions were studied. Final conclusions about the choice of convolutional network models for the road surface feature detection and other concerns are included at the end of the work.

Keywords: neural networks, convolutional neural networks, object detection, road surface feature detection, computer vision, deep machine learning frameworks.

# SATURS

Apzīmējumu saraksts.....	5
Ievads.....	6
1. Mākslīgie neironu tīkli .....	9
1.1. Mākslīgo neironu tīklu vēsture .....	9
1.2. Neironu tīkla uzbūve .....	10
1.3. Neironu tīkla aktivitātes funkcijas .....	11
1.3.1. Sigmoidālās aktivitātes funkcijas .....	12
1.3.2. ReLU aktivitātes funkcijas .....	14
1.4. Neironu tīklu veidi un arhitektūras .....	15
1.5. Neironu tīklu apmācības metodes.....	16
1.5.1. Uzraudzītā apmācīšanās metode.....	17
1.5.2. Neuzraudzītā apmācīšanās metode .....	17
1.5.3. Pastiprinātas apmācīšanās metode.....	18
1.5.4. Zuduma funkcija.....	18
2. Konvolūciju neironu tīkli .....	19
2.1. Konvolūciju neironu tīkla slāņi .....	20
2.1.1. Konvolūciju slānis .....	21
2.1.2. Kopošanas slānis .....	22
2.1.3. Pilnās sasaistes slānis .....	23
2.2. Konvolūciju neironu tīklu veidi.....	23
2.2.1. R-CNN.....	23
2.2.2. Fast R-CNN .....	25
2.2.3. Faster R-CNN .....	26
2.2.4. Mask R-CNN.....	27
2.2.5. Single Shot Detectors .....	28
2.3. Secinājumi .....	30
3. Dziļās mašīnmācīšanās ietvari.....	31
3.1. TensorFlow .....	31
3.2. Caffē un Caffē2 .....	32
3.3. Torch un PyTorch.....	32
3.4. Secinājumi .....	33
4. Ceļa seguma iezīmju noteikšanas problēmas analīze .....	34

4.1. Problēmas analīze .....	34
4.2. Attēlu iezīmju klases .....	35
4.3. Atziņas, kas iegūtas no citiem pētījumiem .....	39
4.4. Rekomendācijas tālākai izpētei un secinājumi .....	40
Rezultāti.....	41
Secinājumi .....	42
Izmantotā literatūra un avoti.....	43
Pielikumi.....	49
1. Pielikums. CNN arhitektūru vispārējās uzbūves kopsavilkums.....	49
2. Pielikums. Konvolūciju neironu tīklu veidu ātrdarbības salīdzinājums atkarībā no attēla izšķirtspējas .....	50
3. Pielikums. Mašīnmācīšanās ietvaru kopskats, 2017. gads .....	50
4. Pielikums. Seguma virsmas novērtēšanas tabula Ontario, Kanāda.....	51

## APZĪMĒJUMU SARAKSTS

- CNN** – Konvolūciju neironu tīkls (angļu: *Convolutional Neural Network*)
- RFID** – Radiofrekvences identifikācija (angļu: *Radio Frequency IDentification*)
- LiDAR** – Gaismas uztveršanas un attāluma mērīšana (angļu: *Light Detection and Ranging*)
- ANN** – Mākslīgais neironu tīkls (angļu: *Artificial neural network*)
- ADALINE** – Adaptīvie, lineārie elementi (angļu: *ADaptive LINear Elements*)
- MADALINE** – Vairāki adaptīvie, lineārie elementi (angļu: *Multiple ADaptive LINear Elements*)
- ReLU** – Kalibrētā, lineārā vienība (angļu: *The Rectified Linear Unit*)
- FNN** – Vienvirziena neironu tīkli (angļu: *Feedforward Neural Networks*)
- MLP** – Daudzslāņu perceptroni (angļu: *Multi-Layer Perceptrons*)
- SVM** – Atbalsta vektora mašīna (angļu: *Support Vector Machine*)
- CIFAR-10** – Anotēta, vairāku miljonu attēlu kopa
- AlexNet** – 8 slāņu konvolūciju neironu tīkla modelis [14]
- R-CNN** – Reģionu bāzētais konvolūciju neironu tīkls (angļu: *Region-based Convolutional Neural Network*)
- RoI** – Interesējošais reģions (angļu: *Region of Interest*)
- YOLO** – Konvolūciju neironu tīkls ar vienreizēju skatīšanos (angļu: *You Look Only Once*)
- SSD** – Viena šāviena detektors (angļu: *Single Shot Detector*)
- VGG-16** – 16 slāņu konvolūciju neironu tīkla modelis, kuru izstrādāja vizuālās ģeometrijas grupa Oksfordas Universitātē (angļu: *Visual Geometry Group - 16*) [23]
- RPN** – Reģionu ieteikšanas tīkls (angļu: *Region Proposal Network*)
- API** – lietojumprogrammas interfeiss (angļu: *Application Programming Interface*)
- PCI** – seguma stāvokļa indekss
- GPS** – globālā pozicionēšanas sistēma (angļu: *Global Positioning System*)

## IEVADS

Pēdējos gados īpaši pastiprināta sabiedrības uzmanība tiek pievērsta mašīnmācīšanās, mākslīgā intelekta un neironu tīklu datorzinātnes apakšnozarēm un tas ir pateicoties pēdējo gadu sasniegumiem šajās nozarēs, kas ir ļāvuši datoriem ar daudz augstāku precizitāti un veiktspēju izpildīt dažādus datorredzes uzdevumus, kā piemēram, mūsdienu, patērētāju klases video kartes var salīdzinoši īsā laikā apmācīt sarežģītu *CNN* modeli izmantojot *ImageNet* attēlu datubāzi [1], kas paredzētā neironu tīklu apmācībai. Skaitļošanas resursu pieejamība šodien dod iespēju jau komerciālā un produktu līmenī risināt objektu un tēlu atpazīšanu rastra grafikas elementos – video un attēlos.

Sasniegumi šajās jomās ir radījuši iespēju un telpu jauniem un inovatīviem produktiem un pakalpojumiem, kā piemēram, *Amazon Go*, kas ir jauna veida sadzīves preču veikals, kurā ir ļoti neliels skaits darbinieku, kas nemaz nav tieši saistīti ar pircēju un veikala apmeklētāju apkalpošanu, jo tieši tas notiek automātiski, t.i., veikalā nav kases vai pārdevēju. Pircēji savu klātbūtni veikalā pie ienākšanas reģistrē ar viedtelefonā uzstādītu *Amazon Go* lietojumprogrammu, kas pircēju sasaista ar viņa *Amazon* kontu un attiecīgo maksājumu karti, kas nodrošina preču uzskaiti un norēķināšanos, paša preču iegāde un iegādāto preču atsekošana notiek automātiski. Veikalā ir izvietots liels skaits dažādu sensoru (piem., *RFID*) un video kameru masīvi, kas ļauj iegūt datus par katru pircēju – kādus produktus viņš paņēms vai novieto atpakaļ vietā, to visu izdara sensoru tīkls, kura pamatā ir datorredzes un mašīnmācīšanās tehnoloģijas, kas ļauj visam šim procesam reāllaikā izsekot no sākuma līdz beigām ar augstu precizitāti. [2]

Pēdējos gados ir arī aktualizējusies un savā mērā arī realizēta pašbraucošo automobiļu ideja un vieni no šīs idejas komerciālajiem pionieriem ir ASV elektrotransportlīdzekļu ražošanas uzņēmums *Tesla* ar automašīnu produktu sēriju *Model*, kur ir iespējams uzstādīt autopilota funkcionalitāti, kuru nodrošina transportlīdzekļa korpusā uzstādītie sensori (*LiDAR*, ultraskaņas starotāji un uztvērēji, u.c.), kā arī video kameras [3]. Visu šo sensoru kopa ļauj transportlīdzeklim “redzēt” apkārtni un spēt rīkoties attiecīgās situācijās tā kā to darītu cilvēks vai pat vēl efektīvāk, tomēr praktiski šīm tehnoloģijām un ierīcēm ir zināmas robežas, trūkumi un problēmas, kuru atrisināšana nebūt nav triviāla.

Kā piemēru var minēt, 2016. gadā notikušo, pirmo nāvējošo ceļu satiksmes negadījumu, kurā bija iesaistīts *Tesla* transportlīdzeklis, kas tika vadīts izmantojot autopilota funkcionalitāti. Transportlīdzeklī uzstādītā autopilota programmatūra nespēja atšķirt gaišu traktoru, kas perpendikulāri šķērsoja brauktuvi no gaišām debesīm, kā rezultātā netika automātiski

iedarbinātas transportlīdzekļa bremzes, lai izvairītos no sadursmes. [4] Vēl viens ceļu satiksmes negadījums notika pavisam nesen, 2018. gada martā, kad *Tesla* elektrotransportlīdzeklis darbojoties autopilota režīmā, diennakts tumšajā laikā “neredzēja” cilvēku, kas šķērsoja brauktuvi grūti pārrēdamā vietā, kā rezultātā notika transportlīdzekļa sadursme ar ceļu šķērsojošo cilvēku. [5]

Objektu noteikšana uz brauktuves ir tikai viena no datorredzes sistēmu, pašbraucošajos transportlīdzekļos, daudzajām problēmām – ir nepieciešams arī spēt izšķirt objektus un iezīmes uz ceļa virsmas, kā piemēram, joslu un brauktuvi atdalošās līnijas, gājēju pārejas, pieturvietas, apstāšanās vietas līnijas, brauktuves seguma bojājumus (plaisas, bedres, u.tml.) un citus brauktuves seguma elementus.

Šī problēma ir vēl joprojām aktuāla un tiešsaistes plašsaziņu līdzekļos parādās video fragmenti ar *Tesla* automobili, kas tiek vadīts autopilota režīmā, nespēj pareizi paredzēt kā turpināsies brauktuves josla pie remontdarbu veikšanas vietām un tiltu estakādes nobrauktuvēm vai uzbrauktuvēm, rezultātā transportlīdzeklis turpinot un nemainot kustības virzienu nonāk situācijā, kur vadītājs ir spiests pašrocīgi strauji bremzēt vai mainīt braukšanas virzienu, jo braukšanas trajektorija saduras ar ceļa brauktuvi sadalošo barjeru. [6]

Ir redzams, ka pēdējo gadu sasniegumi datoru veikspējas uzlabošanā, kā arī mašīnmācīšanās un neironu tīklu topoloģijas un algoritmu izstrādē ir bijis vēra ņemams progress, kas ir ļāvis šo produktu un pakalpojumu nonākšanai sabiedrībā, taču arī ir redzams, ka vēl joprojām pastāv nepilnības. Viena no jomām, kur datorredzes potenciāls varētu sevi pierādīt ir ceļu uzturēšana. Pēdējos piecos gados Latvijas valsts autoceļu ikdienas uzturēšanai, gadā, vidēji tiek tērēti 75,6 milj. EUR, no kuriem, vidēji, 10,5 milj. EUR tiek izlietoti ceļu kopšanas, apsekošanas un citiem darbiem. Autoceļu vizuālā apsekošana katru gadu tiek veikta pēc vienotas metodikas. Iegūtais ceļu segumu novērtējums ir viens no kritērijiem rekonstrukcijas un periodiskās uzturēšanas programmu izveidei. 2016. gadā 44% melnā segumu un 42.6% grants segumu ceļu bija sliktā vai ļoti sliktā stāvoklī [7]. Viens no ceļu uzturēšanas izmaksu samazinājuma priekšnosacījumiem ir savlaicīga un pareizi izvēlēta bojāto un nolietoto ceļu labošana, lai nebūtu jāveic bojātā ceļa posma pārbūve, kas potenciāli izmaksā daudz vairāk kā regulāra ceļa uzkopšana, kā arī pareiza ceļa seguma tipa un materiāla izvēle attiecībā pret ceļa noslodzi, meteoroloģiskajiem apstākļiem, ģeogrāfiskā novietojuma un ģeoloģiskajām īpatnībām.

Darba mērķis ir apskatīt un izpētīt esošos datorredzes un mašīnmācīšanās risinājumus, kuriem ir izmantošanas potenciāls ceļa seguma iezīmju noteikšanā, kas ļautu atvieglot vai daļēji automatizēt ceļu uzturēšanas darbus.

Pētījuma laikā ir plānots:

- Izpētīt neironu tīklu darbības principus un attīstīšanās vēsturi;
- Izpētīt konvolūciju neironu tīklu uzbūvi, jaunākos veidus un darbības principus;
- Apskatīt dažas no pieejamajām mašīnmācīšanās bibliotēkām;
- Izpētīt un gūt atziņas no pēdējos gados veiktajiem pētījumiem, kas ir veikti šīs tēmas sakarā, potenciāli iezīmējot apzinātos problēmu, apgrūtinājumu un citu vai citu alternatīvo risinājumu apgabalus;
- Izpētīt ar ceļu uzturēšanu saistītus informācijas avotus un pētījumus;
- Apkopot pētījuma rezultātus un izdarīt secinājumus, dodot rekomendācijas praktiska eksperimenta veikšanai, piemēram konvolūciju neironu tīkla veida izvēlē, mašīnmācīšanās ietvara izvēlē kā arī tālākā pētniecības darba turpināšanā;

Faktoloģiskais materiāls, tika iegūts no publiski pieejamām, tīmekļa vietnēm, grāmatām, zinātniskajiem rakstiem un citiem tīmekļa resursiem.

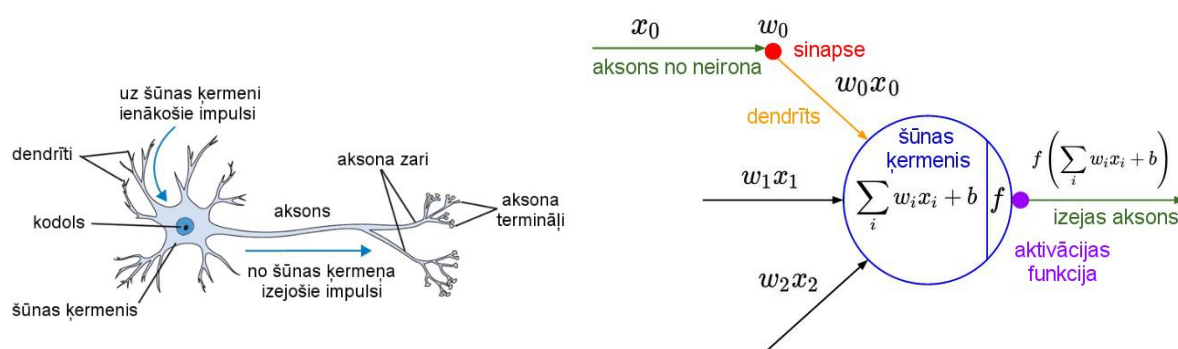
Darbs ir strukturizēts nodaļās un apakšnodaļās. Darba sākumā ir pirmā nodaļa, kur tiek apskatītas mākslīgo neironu tīklu attīstības vēsture un darbības principi. Darba otrajā nodaļā tiek apskatīti konvolūciju neironu tīklu un to atvasināto neironu tīklu darbības principi un struktūra. Darba trešajā nodaļā tiek īsumā apskatītas un aprakstītas pieejamās mašīnmācīšanas bibliotēkas to darbību principi kā arī rekomendācijas bibliotēku izvēlē balstoties uz pieejamo informāciju. Ceturtajā darba nodaļā tiek apskatīti ar ceļu seguma iezīmju noteikšanu un ceļa seguma uzturēšanu saistītās problēmas, pētniecības darbi un informācijas avoti.

Pētniecības darba beigu daļa ir veltīta izvīzīto mērķu un to rezultātu apkopošanai, secinājumiem un potenciālās tālākās izpētes virzienu apzināšanai kā arī rekomendāciju sniegšanai neironu tīklu izveidē, klasifikatoru izvēlē. Tiek apzinātas potenciālās problēmas ar kādām varētu saskarties turpinot pētījumu, tajā skaitā praktiskos eksperimentus, šīs tēmas sakarā balstoties uz pasaulē iepriekš veiktiem pētījumiem.

# 1. MĀSKLĪGIE NEIRONU TĪKLI

Šajā nodaļā tiks virspusēji apskatīta ANN (angļu: *Artificial Neural Networks*), jeb mākslīgo neironu tīklu uzbūve un vēsturiskā attīstība līdz mūsdienām. Tiks aprakstītas mākslīgo neironu tīklu funkcijas, neironu tīklu veidi un izmantošanas piemēri.

Mākslīgais neironu tīkls pēc būtības ir matemātisks modelis, kas ir veidots iedvesmojoties no bioloģisko neironu uzbūves un darbības. Cilvēka nervu sistēmā ir aptuveni 86 miljardi neironu un tie ir savstarpēji savienoti ar aptuveni  $10^{14} - 10^{15}$  nervu sinapšu. Mākslīgā neirona matemātiskais modelis, skatīt 1.1. att. (*pa labi*), pēc būtības līdzinās dabiskā, bioloģiskā neironu uzbūvei, skatīt 1.1. att. (*pa kreisi*). [8]



1.1. att. Dabīgā un mākslīgā neirona modeļi [8]

Ir svarīgi paturēt prātā to, ka šī matemātiskā neirona modeļa analogija ir tikai virspusēja, piemēram, bioloģiskajos neironu tīklos ir vairāki neironu veidi, katrs ar savām īpašībām un funkcijām. Dendrīti bioloģiskajos neironos izpilda kompleksas, nelineāras darbības, neironu sinapses nav vienkārši svāri noteiktu darbību stimulācijai, tās ir kompleksas, nelineāras un dinamiskas sistēmas – ir zināms, ka noteiktām laikspraugām ir liela nozīmē daudzu bioloģisko sistēmu darbībā un stimulēšanā, kā rezultātā bioloģisko neironu darbības simulēšana ar matemātiskajiem modeļiem ir, salīdzinoši, stipri vienkāršots bioloģiskā neirona modelis. [9]

## 1.1. Mākslīgo neironu tīklu vēsture

Mākslīgo neironu tīklu idejas pirmsākumi ir meklējumi jau tālajos 1940. gados, kad neurofiziologs Varens Makkaloks (angļu: *Warren McCulloch*) un Valters Pitts (angļu: *Walter Pitts*) uzrakstīja darbu par to kā varētu darboties bioloģiskie neironi. Lai aprakstītu to, kā darbojas smadzeņu neironi, viņi uzmodelēja vienkāršu neironu tīklu izmantojot elektriskās ķēdes un elektriskos elementus. Svarīgs moments agrīnajā neironu tīklu attīstībā bija Frenka Rozenblata (angļu: *Frank Rosenblatt*) un viņa līdzstrādnieku izstrādātais t.s. perceptorns

Masačūsetsas tehnoloģiskajā institūtā (MIT) 1958. gadā. [10] 1959. gadā, Bernards Vidrovs (angļu: *Bernard Widrow*) un Marcians Hofs (angļu: *Marcian Hoff*) no Stenfordas izstrādāja pirmos neironu tīkla modeļus *ADALIN* un *MADALINE*. *ADALINE* tika izstrādāts, lai atpazītu binārās iezīmes tā, lai lasot bitu straumi no telefonu sakaru līnijas tas spētu paredzēt nākamo bitu. *MADALINE* bija pirmais neironu tīkla modelis, kam bija pielietojums praksē – izmantojot adaptīvo filtru, kas no telefonu sakaru līnijām izņēma atbalsi. [11]

Neironu tīklu attīstību stipri aizkavēja 1969. gadā izdotā Minska un Peiparta (angļu: *Minsky and Papert*) izdotā grāmata, kurā tika pierādīts, ka perceptrona modelis nespēja atrisināt tik vienkāršas problēmas kā t.s. XOR problēma, kur nu vēl sarežģītākas. Līdz ar šo daudzos neironu tīklu projektos tika samazināts vai noņemts finansējums, un pētījumus turpināja tikai entuziasti. [10]

1982. gadā interese mākslīgo neironu tīklu jomā atdzīvojās, kad Džons Hopfields (angļu: *John Hopfield*) no Kalifornijas Tehnoloģiju institūta uzrakstīja papīru par lietderīgākiem neironu tīkliem, kas izmanto divvirzienu saiknes starp neironiem, līdz šim brīdim mākslīgajos neironu tīklos tika izmantotas vienvirziena saites starp neironiem. 1986. gadā parādījās ziņas par pirmajiem vairākslāņu neironu tīkliem, trīs neatkarīgas pētnieku grupas, vienā no kurām bija Stenfordas Universitātes psiholoģijas nodaļas bijušais biedrs Deivids Rumelhārts (angļu: *David Rumelhart*), kurš nāca klajā ar atmuguriskās izplatīšanās (angļu: *back propagation*) neironu tīkliem, kas ļāva iezīmju noteikšanas kļūdām izdalīties pa visu neironu tīklu uzlabojot noteikšanas precizitāti, taču šie tīkli bija pazīstami kā lēni apmācāmi, jo to apmācībai bija nepieciešami vairāki tūkstoši iterāciju. [12]

## 1.2. Neironu tīkla uzbūve

Mākslīgais neirons ir informācijas apstrādes vienība, kas ir neironu tīklu darbības fundamentāla sastāvdaļa. Nodaļas sākumā parādītais mākslīgā neirona modelis, skatīt 1.1 att (*pa kreisi*), ir pamata sastāvdaļa lieliem neironu tīkliem. Mākslīgais neirons sastāv no trijām pamata elementiem:

1. Kopa ar sinapsēm, jeb savienojumu saiknēm, kur katra saikne tiek raksturota ar tās attiecīgo svaru vai stiprumu. Konkrētāk signāls  $x_i$  pie ieejas sinapses  $i$ , kas ir savienota ar neironu  $k$  tiek pareizināts ar saiknes svaru  $w_k$ .
2. Summēšanas elements jeb funkcija, kas sasummē visas ieejas saikņu svērtās vērtības un izrēķina vienu vērtību, kas tālāk tiek padota vai nu noslieces parametra summēšanai un tad aktivitātes funkcijai vai pa tiešo aktivitātes funkcijai.

3. Aktivitātes funkcija, kas limitē neirona izejas vērtības noteiktā vērtību apgabalā un veidā atkarībā no aktivitātes funkcijas. Šī funkcija ir svarīgākā neirona darbības raksturotāja, kas var norādīt uz neirona spēju risināt noteiktas sarežģītības uzdevumus.

Tipiski mākslīgo neironu izejas vērtības tiek ierobežotas intervālā no  $[0,1]$  vai  $[-1,1]$ , kas atbilst varbūtiskajiem matemātiskajiem neironu modeļiem, piemēram, plašākā, vienkāršotākā skatījumā, ja neironu kopa konkrētam ieejas tēlam vai iezīmei rezultātā izvada vērtību, kas ir vienāda vai tuvinās 1, tad tiek pieņemts, ka šis tēls vai iezīme atbilst tādas klases vai superklases iezīmei ar kādu šī neironu kopa tika apmācīta ar iegūto varbūtības vērtību.

Attēlā 1.1 (*pa labi*) attēlotais mākslīgā neirona modelis neatspoguļo vēl vienu mākslīgo neironu tīklu īpašību, kur pēc summēšanas funkcijas var tikt ierēķināta arī neirona nosliece  $b_k$  atkarībā no neironu tīkla arhitektūras un uzbūves. Noslieces iekļaušana rada pozitīvu vai negatīvu summēšanas funkcijas efektu atkarībā no noslieces  $b_k$  parametra.

Matemātiski mākslīgo neironu  $k$  un tā summēšanas funkciju var raksturot ar diviem matemātiskajiem vienādojumiem (1.1 un 1.2):

$$u_k = \sum_{i=1}^m w_{ki} x_i \quad (1.1)$$

un

$$y_k = \varphi(u_k + b_k) \quad (1.2)$$

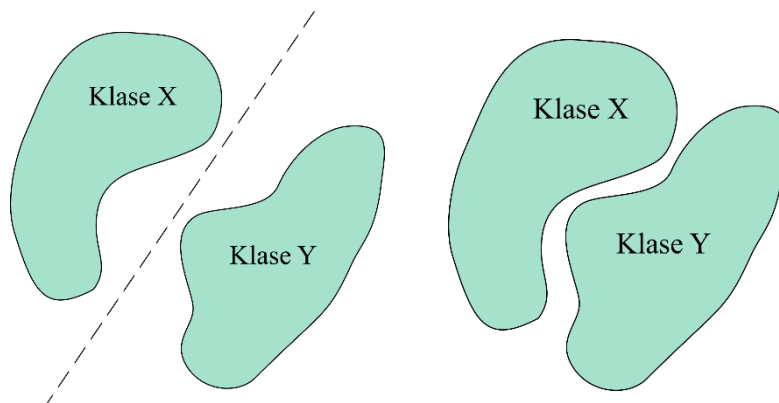
,kur  $x_1, x_2, \dots, x_m$  ir ieejas signāli;  $w_{k1}, w_{k2}, \dots, w_{km}$  ir attiecīgie sinapšu svāri neironam  $k$ ;  $u_k$  ir lineārā kombinētāja izejas vērtībā attiecībā no ieejas vērtībām;  $b_k$  ir noslieces parametrs, kas transformē  $u_k$  izejas vērtību. [10,13]

### 1.3. Neironu tīkla aktivitātes funkcijas

Neironu izejas vērtība lielā mērā ir atkarīga no aktivitātes funkcijas, pamatā ir vairāki, visbiežāk sastopamo aktivitāšu funkcijas:

- Lineāra aktivitātes funkcija;
- Sliedzveida aktivitātes funkcija;
- Lineāra aktivitātes funkcija, kas ierobežota intervālā;
- Loģistiskā (sigmoidālās) aktivitātes funkcija (viena no populārākajām);
- Hiperboliskā tangensa aktivitātes funkcija;
- ReLU aktivitātes funkcija (viena no vislabāk strādājošajām);

Aktivitātes funkcija ir svarīgākā neirona darbības raksturotāja, jo tā norāda uz neirona spēju risināt noteiktas sarežģītības uzdevumus, piemēram, lineāra aktivitātes funkcija spētu labi tikt galā ar lineāra rakstura problēmām, piemēram, skatīt 1.2 att., kur ir attēlotas divas dažādas klases, X un Y, kā piemērs lineārai, 1.2 att. (*pa kreisi*) vai nelineārai, 1.2. att. (*pa labi*) tēlu iedalīšanai atbilstošajās klasēs, sadalīšanas problēmai.

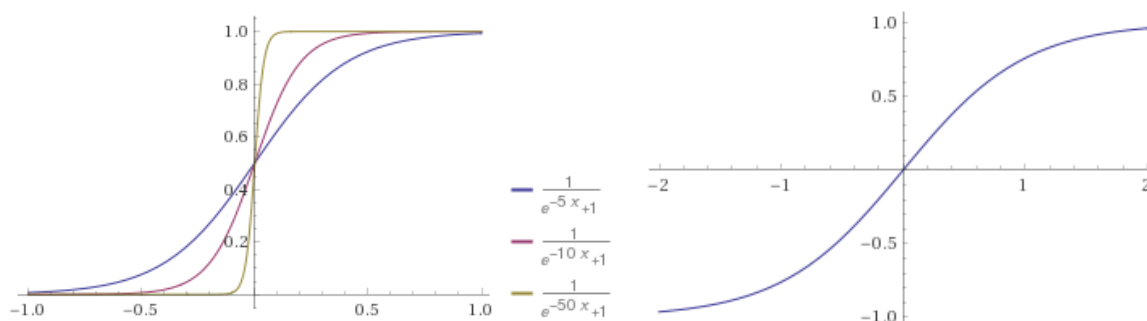


1.2. att. Lineāras un nelineāras tēlu sadalīšanas problēmas piemērs [13]

Katrai mākslīgo neironu tīklu aktivitātes funkcija ir savi plusi un mīnusi, taču objektu un tēlu atpazīšanai attēlu uzdevumos visbiežāk tiek lietotas loģistiskās un *ReLU* aktivitātes funkcijas un to atvasinājumi, kas tiks apskatīts detalizētāk šajā apakšnodaļā.

### 1.3.1. Sigmoidālās aktivitātes funkcijas

Sigmoidālās jeb loģistiskās funkcijas pēc formas veido S-veida grafiku. Sigmoidālās aktivitātes funkciju izvēle neironu tīklos ir ar bioloģisku pamatojumu, jo tās simulē bioloģiskā neirona stimulu organismā, tās ir arī ērtas, jo ļauj ierobežot neirona izejas vērtību noteiktā intervālā, tādā veidā nodrošinot normalizācijas efektu, kas ir praksē ir ļoti vērtīgs. Sigmoidālās funkcijas ir nelineāras, bet, atšķirībā no sliekšņveida funkcijām – arī atvasināmas, kas arī ir svarīga īpašība praksē, skatīt 1.3. att. [10, 13]



1.3. att. Lineāras un nelineāras tēlu sadalīšanas problēmas piemērs [13]

Viena no populārākajām aktivitātes funkcijām ir t.s. loģistiskā aktivitātes funkcija, kas ir matemātiski aprakstāma ar vienādojumu (1.3):

$$\sigma(x) = \frac{1}{1 + e^{-\left(\frac{1}{g}x\right)}} \quad (1.3)$$

,kur  $g$  ir līknes slīpuma koeficients un hiperboliskā tangensa funkcija, kas ir matemātiski aprakstāma ar vienādojumu (1.4):

$$\sigma(x) = \frac{(e^x) - (e^{-x})}{(e^x) + (e^{-x})} \quad (1.4)$$

Abu sigmoidālo funkciju vērtīga īpašība ir tā, ka to atvasinājums ir izsakāms ar pašas funkcijas vērtību, kas ir nozīmīgs faktors neironu tīklu modeļa – perceptrona apmācībā. [10]

Reālās pasaules uzdevumos retāk tiek izmantota loģistiskās aktivitātes funkcija, formula (1.3), jo tai ir divi būtiski trūkumi, kas negatīvi ietekmē mākslīgo neironu tīklu darbību un apmācību reālajā pasaulē.

Pirmkārt, loģistiskās aktivitātes funkcijas var piesātināt un padarīt nelietojamu funkcijas gradientu. Ļoti nevēlama īpašība loģistisko aktivitātšu funkciju klasei ir tā, ka, ja mākslīgais neirons tiek stimulēts tuvu robežvērtībām 0 vai 1, tad funkcijas gradients šajos reģionos ir līdzvērtīgs nullei, kas pie at mugurniskās izplatīšanās darbības šo vēl vairāk efektu pastiprina rezultātā lokālais gradients ir ļoti neliels, kas praktiski sabojā neirona spēju darboties un signalizēt. Papildus šim ir arī piesardzīgi jāizvēlas neirona savienojuma sākotnējie svāri, ja svāru vērtība ir par lielu, tad neirons var tikt pārsātināts ar signālu un zaudēt mācīšanās spēju.

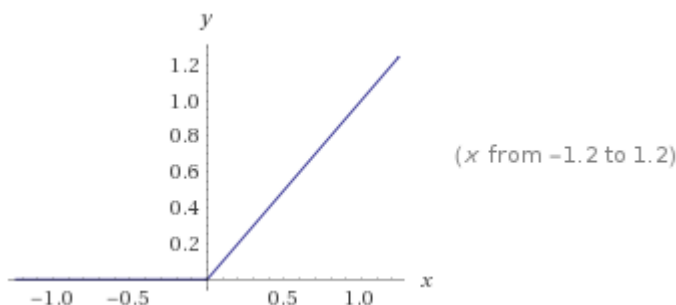
Otrkārt, loģistiskās aktivitātes funkcijas izejas vērtības nav nulles centrētas. Arī šī funkciju īpašība nav pārāk vēlama mākslīgajos neironu tīklos, jo pēc primārās datu apstrādes, tālākiem neironu slāņiem var tikt padotas vērtības, kas nav nulles centrētas, rezultātā var būt gadījumi, kad neironā ienākošie dati ir vienmēr pozitīvi, kas nozīmē, ka funkcijas gradients atkarībā no noslieces faktora var atsevišķos gadījumos vienmēr būt pozitīvs vai vienmēr negatīvs, taču šī problēma nav tik būtiska kā pirmā, jo neironi savstarpēji var mazināt šī efekta spēku.

Atšķirībā no loģistiskās funkcijas, formula (1.3), hiperboliskā tangensa funkcija, formula (1.4) ir nulles centrēta, kas nozīmē, ka augstāk minētās problēmas vairs nav tik aktuālas praktiskajos uzdevumos tajā pat laikā saglabājot loģistiskās aktivitātes funkcijas labās īpašības, kas simulē bioloģiskā neirona darbību. [8]

### 1.3.2. ReLU aktivitātes funkcijas

Pēdējo gadu laikā ļoti populāra ir kļuvusi *ReLU* aktivitātes funkcija, skatīt 1.4 att. un formulu (1.5).

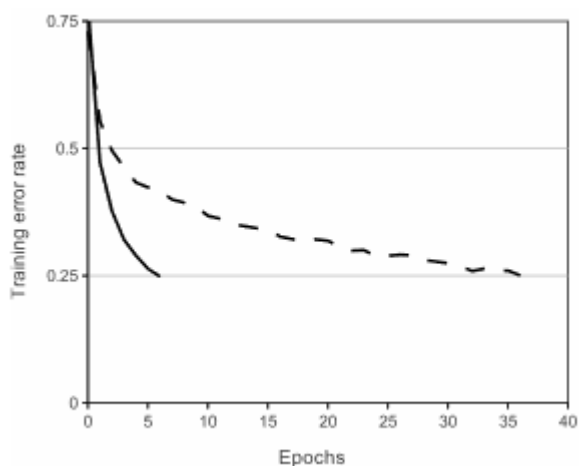
$$f(x) = \max(0, x) \quad (1.5)$$



1.4. att. ReLU aktivācijas funkcijas piemērs [8]

Šīs aktivitātes funkcijas darbība ir līdzīga sliekšņveida aktivitātes funkciju klasē esošajām funkcijām, taču arī šai aktivitātes funkciju klasei ir savas priekšrocības un trūkumi.

Pirmkārt, viena no priekšrocībām tika atklāta pētījumā “ImageNet Classification with Deep Convolutional Neural Networks”, kur tika noskaidrots, ka izmantojot *ReLU* aktivitātes funkciju konvolūciju neironu tīkla apmācīšanā attēlu klasificēšanā – funkcijas gradienta konverģences uz tās zemāko punktu ātrums palielinājās pat par sešām reizēm salīdzinājumā ar sigmoidālo vai hiperboliskā tangensa aktivitātes funkcijām, skatīt 1.5. att. Tas ir izskaidrojams ar tās nelineāro dabu, neironus nepiesātinošo īpašību un ātrdarbību. [8, 14]



1.5. att. ReLU ātrdarbības salīdzinājums ar sigmoidālām aktivitātes funkcijām [14]

Otrkārt, salīdzinājumā ar hiperboliskā tangensa un sigmoidālām aktivitātes funkcijām, kas ietver dārgas skaitļošanas operācijas (eksponentu aprēķini), *ReLU* aktivitātes funkcija var tikt implementēta ierobežojot aktivāciju matricu vērtības pie nulles.

Diemžēl arī *ReLU* ir savi trūkumi un neironi, apmācības procesos, var kļūt piesātināti un nereaģēt uz apmācību, piemēram, nepareizi izvēlēti neironu tīkla apmācības parametri var novest pat pie 40% neironu, kas nepilda savu funkciju un nereaģē uz nevienu no apmācības datu kopā esošajiem piemēriem. Ir vēl divas vispārzināmas aktivitātes funkcijas, kas risina neironu piesātinājuma problēmu un tās ir – *Leaky ReLU* un *Maxout*. *Leaky ReLU* attiecīgi neironu elementu piesātinājuma iespējamību novērš ļaujot aktivitātes funkcijas rezultātam būt nelielam, negatīvam skaitlim, ko var piekoriģēt ar konstanti. [8]

*Maxout* izmanto citu pieeju – neironu funkcijas pamatā ir formula (1.6.):

$$\sigma(x) = \max(w_1^T x + b_1, w_2^T x + b_2) \quad (1.6)$$

Šāda pieeja ļauj neirona elementam izmantot visas labās *ReLU* un *Leaky ReLU* īpašības tajā pat laikā izvairoties no neironu piesātināšanas problēmas taču *Maxout* neironam ir divreiz vairāk parametru priekš katra neirona tādā veidā veidojot lielu skaitu ar kopējiem parametriem neironu tīklā. [8, 15]

#### 1.4. Neironu tīklu veidi un arhitektūras

Mākslīgie neironu tīkli sastāv no neironu elementiem, kuri atkarībā no neironu tīklu uzbūves un funkcijas var tikt izvietoti pēc dažādām topoloģijām. Neironu tīklus pamatā var iedalīt trīs kategorijās:

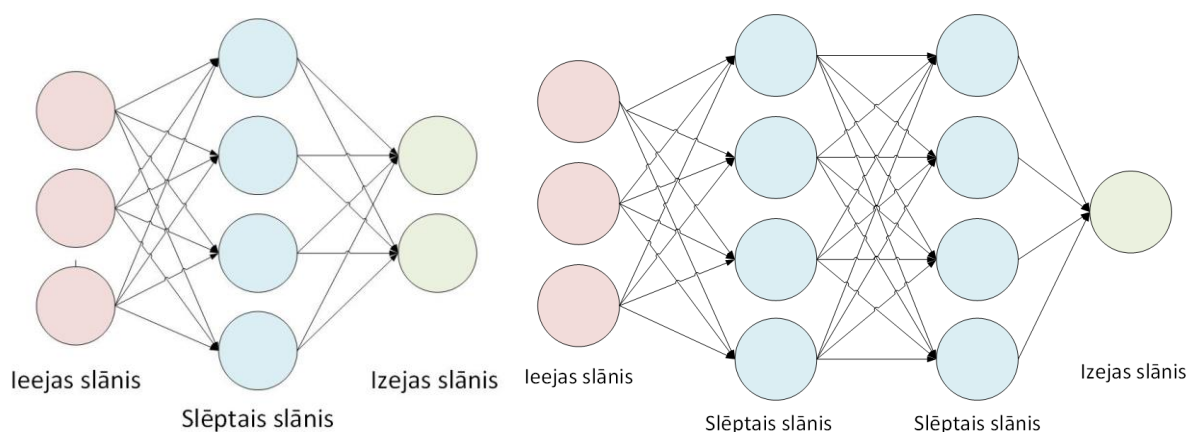
1. Vienslāņa vienvirziena neironu tīkli;
2. Daudzslāņu vienvirziena neironu tīkli;
3. Atgriezeniskās saites neironu tīkli;

Šajā darbā tiek apskatīti konvolūciju neironu tīkli, kas ir ļoti populāri objektu un tēlu atpazīšanas datorredzes uzdevumos un šo tīklu pamatā ir vienvirziena neironu tīkli, tāpēc sīkāk tiks apskatīta šo tīklu uzbūve. [13]

Vienslāņa vienvirziena neironu tīkli tiek organizēti slāņos – ieejas un izejas slānī. Neironi ieejas slānī ir vienvirziena veidā pilni savienoti ar izejas slāņa neironiem un tiem nav atgriezeniskās saites. Ieejas slāņa neironos netiek veiktas nekādas skaitļošnas darbības tādēļ ieejas slānis netiek skaitītes pie kopējā slāņu skaita, ka rezultātā arī ir radies tāds nosaukums – vienslāņa vienvirziena neironu tīkls. Šis ir pats vienkāršākais vienvirzienu neironu tīklu veids.

Daudzslāņu vienvirziena neironu tīkli no vienslāņu, vienvirziena neironu tīkliem atšķiras ar to, ka starp ieejas un izejas slāni ir vismaz viens vairāki slēptie neironu slāņi. Šos slāņus sauc par slēptiem, jo tie nav tiešā veidā redzami no ieejas un izejas vērtībām. Šo slēpto slāņu loma neironu tīklā ir veikt globālāku statistisko iezīmju noteikšanu starp ieejas un izejas slāņiem. Tipiski savienojumi tiek veidoti starp blakus esošajiem slāņiem. [8, 10, 13]

Neironu tīkli var tikt modelēti, kā kopa ar savstarpēji savienotiem elementiem jeb grafs, kurā nav ciklošanās. Daļa neironu var būt kā ieeja citiem neironiem, taču ciklošanās vienvirziena neironu tīklos, jeb *FNN* (angļu: *Feedforward Neural Networks*) – nav atļauta. Tā vietā, lai neironi būtu savstarpēji savienoti vienā, lielā amorfā kaudzē, tie tiek organizēti vairākās kārtās. Viens no visizplatītākajiem neironu tīklu kārtas veidiem ir pilni savstarpēji savienoto neironu kārtas, kur neironu starp divām blakus esošām kārtām ir pilnīgi, savstarpēji savienoti, t.i. vienas kārtas visu neironu izejas vērtības savienojas ar visiem otras kārtas neironiem kā ieejas vērtības, skatīt 1.6. att (*pa kreisi*).



1.6. att. Vairākslāņu neironu tīklu populārākās topoloģijas [14]

Daudzslāņu vienvirziena neironu tīkliem ir noteikts veids kā tie tiek nosaukti, piemēram, 1.6. att. (*pa labi*) redzamais neironu tīkls ir trīs slāņu, vienvirziena neironu tīkls. Cits piemērs, kā tiek veidots daudzslāņu vienvirziena neironu tīkla nosaukums –  $m$  skaits ieejas neironu,  $h_1$  neironu skaits pirmajā slēptajā slānī,  $h_2$  neironu skaits otrajā slēptajā slānī un  $q$  neironu skaits izejas slānī. Šādas konfigurācijas tīklu sauc par  $m - h_1 - h_2 - q$  neironu tīklu.

## 1.5. Neironu tīklu apmācības metodes

Viena no neironu tīklu galvenajām īpašībām ir to spēja apgūt kādas noteiktas problēmas risināšanu, piemēram, objektu vai tēlu atpazīšanu attēlos vai rekomendācijas sarakstu veidošanu balstoties uz iepriekš zināmiem datiem, u.c. līdzīgas problēmas. Lai neironu tīkli šīs problēmas varētu atrisināt, tos ir nepieciešams apmācīt, jeb neironiem dotajā neironu tīklā ir jāspēj reaģēt uz ievaddatiem un gala rezultātā izdodot vērtībās kas atbilst problēmas risinājumam. Neironu tīkla apmācības mērķis ir minimizēt kļūdu, jeb zuduma funkcijas (angļu: *loss function*) rezultātu, tāpēc ir svarīgi izvēlēties problēmai un izvēlētajam neironu tīkla uzdevumam atbilstošu zuduma funkciju vai arī to definēt pašam.

Neironu tīklu apmācības pamatā ir trīs galvenās apmācības metodes, kuras tiek izvēlētas

atkarībā no ārējās vides (apmācības paraugi, papildus informācija, algoritms, u.c.) ietekmes uz neironu tīklu svaru un citu parametru izmaiņu un tās ir:

1. Uzraudzītā apmācīšanās metode;
2. Neuzraudzītā apmācīšanās metode;
3. Pastiprinātas apmācīšanās metode; [13]

### **1.5.1. Uzraudzītā apmācīšanās metode**

Uzraudzīto apmācīšanas metodi pēc analogijas var salīdzināt kā mācīšanos ar “skolotāju”, kur neironu tīklu gadījumā skolotājs ir tas, kas pārzina vidi un tam ir zināšanas un piemēri par to, pie kādiem ieejas datiem ir sagaidāmie izejas dati, taču informācija par vidi neironu tīklam nav pieejama. Pieņemot, ka gan skolotājs, gan neironu tīkls saņem apmācības vektoru (t.i., piemēru), kas ir ņemts no vienas un tās pašas vides. Pēc būtības skolotājs pēc iebūvētās zināšanu bāzes spēj neironu tīklam padot sagaidāmo rezultātu no šī ieejas vektora. Neironu tīkla optimālā atbilde ir skolotāja sagaidāmais rezultāts, taču tas ir tikai ideālajā gadījumā vai gadījuma, kad neironu tīkls ir pietiekami apmācīts. Neironu tīkls gala rezultātā atgriezīs izrēķinātās vērtības no ieejas vektora, kuras pēc tam tiek apstrādās un tiek noteikts kļūdas koeficients, kas kā kļūdas signāls (nosliece) tiek padots atpakaļ uz neironu tīklu, kur attiecīgi tiek pamainīt neironu svāri jeb to savienojumu stiprums.

Šādā veidā apmācot neironu tīklu tiek panākts, ka neironu tīkls imitē skolotāja darbību un uz attiecīgajiem apmācības piemēriem, ideālajā gadījumā, ar lielu varbūtību, atgriezīs rezultātu, kādu to sagaida skolotājs. Šajā iteratīvajā procesā vides informācija caur “skolotāju” tiek nodota neironu tīklam, kā rezultātā, kad neironu tīkls ir izmainījis neironu svarus un tas dotajiem piemēriem atbild kā tiek sagaidīts, tad var uzskatīts, ka neironu tīkls ir apmācīts. Šīs zināšanas par vidi ir saglabātas neironu svaros jeb neironu tīklu izpratnē – neirona tīkla ilgtermiņa atmiņā.

Šī procesa rezultātā apmācītajam neironu tīklam var tikt doti uzdevumi, kas atbilst definētajiem apmācības piemēriem, taču tagad jau no pašas vides – bez skolotāja starpniecības. Šādu uzraudzītās apmācīšanas metodi arī dēvē par apmācību ar kļūdu labošanu un atgriezenisko saiti. [10, 13]

### **1.5.2. Neuzraudzītā apmācīšanās metode**

Neirona tīkla apmācīšanu bez “skolotāja” sauc par neuzraudzīto apmācīšanas metodi – šajā gadījumā tīklam netiek sniegti marķēti piemēri ar sagaidāmo rezultātu. Neuzraudzītās apmācīšanas metodes pamatā ir neironu tīkla apmācīšana ar vidē sastopamajiem datiem ļaujot

neironu tīklam pēc vairākām iterācijām statistiski atpazīst ievaddatu un tos klasificēt pēc atšķirībām kādas visstiprāk bija ietekmējušas neironu tīklu apmācības procesā, šādā veidā neironu tīkls pats automātiski veic piemēru klasifikāciju jaunās, iepriekš nedefinētās klašu kopās.

Viena no neuzraudzītās apmācīšanās realizācijas formām ir apmācīšana ar konkurenci. Lai šo ilustrētu, pieņemsim, ka tiek izmantots neironu tīkls, kas sastāv no diviem slāņiem – ieejas slāņa un konkurences slāņa. Ieejas slānis saņem visus ieejas datus, taču konkurences slānī esošie neironi savā starpā “konkurē” par to kurš būs spējīgs atbildēt uz iezīmēm ieejas datus visvairāk. Pēc šādas stratēģijas, neirons, kas uz ieejas datiem signalizē par noteiktas iezīmes klātesamību tiek stimulēts, kamēr pārējie ir neaktīvi. [13]

### **1.5.3. Pastiprinātas apmācīšanās metode**

Pastiprinātās apmācīšanās metodē, ieejas un izejas datu kartēšana neirona tīklā tiek veikta izmantojot nepārtrauktu saikni ar vidi. Viena no pastiprinātās apmācīšanās metodes atšķirībām, ir tā, ka apmācības sistēma tiek būvēta ap “kritiķi”, kas pārvērš primāro pastiprināto signālu no vides, augstākas kvalitātes pastiprinātā signālā, ko sauc par heuristiski pastiprināto signālu.

Neironu tīkls pastiprinātās apmācīšanās metodē mācās no aizturētā pastiprinājuma (angļu: *delayed reinforcement*), kas nozīmē, ka neironu tīkls apmācības laikā novēro stimulus no vides (ieejas datiem), kā rezultātā kādā noteiktā brīdī tā svāri tiek mainīti ar heuristiski pastiprināto signālu no “kritiķa”, papildus faktors apmācīšanai ar pastiprinājumu ir tas, ka apmācītās sistēmas izeja tiek interpretēta kā darbība, nevis gala risinājums vai rezultāts.

Apmācīšanās ar pastiprinājumu nereti tiek jaukta ar neuzraudzītās apmācīšanās metodi, piemēram, atšķirībā no apmācības ar skolotāju, nav tieši zināms katras ieejas parauga novērtējums, bet sasniedzamais mērķis ir globālāks – neironu tīkls ne vienmēr labo neoptimālas darbības, tādējādi tiek raksts balanss starp jaunu ievaddatu iezīmju apgūšanu un esošo zināšanu pielietošanu. [8, 10, 13]

### **1.5.4. Zuduma funkcija**

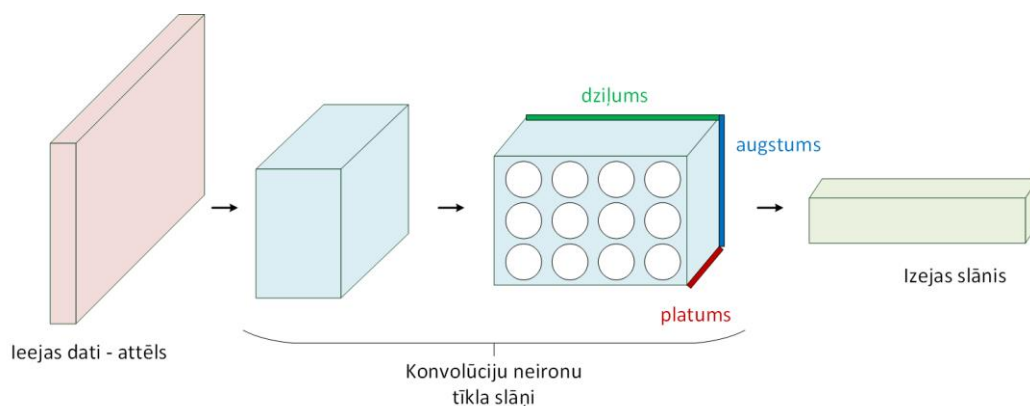
Viens no uzraudzītās apmācīšanās metodes stūrakmeņiem ir neironu tīkla spēja mācīties no kļūdām. Ar zuduma funkcijas palīdzību neironu tīkla izejas vērtības tiek salīdzinātas ar sagaidāmajām, piemēros dotajām, vērtībām. Šādā veidā ir iespējams novērtēt to cik pareizi neironu tīkls veic atbilstošo uzdevumu, kas ļauj arī optimizējot tā darbību. Atbilstošas zuduma funkcijas izvēlē dotajam izdevumam ir viens no galvenajiem kritērijiem pareizā un veiksmīgā neironu tīkla apmācībā. [8, 13]

## 2. KONVOLŪCIJU NEIRONU TĪKLI

Objektu, tēlu un iezīmju noteikšanai attēlos vislabāk piemēroti ir konvolūciju neironu tīkli, kas saistās ar dziļās māšīnmācīšanās apakšnozari. Tie pēc būtības ir mākslīgi neironu tīkli, kas sastāv no savstarpēji savienotiem neironiem, kam ir apmācāmi svaru un noslieces parametri, kur katrs neirons veic skalāro reizinājumu, kam nereti seko nelineārās aktivācijas funkcijas (piem., *ReLU*). Šāds neironu tīkls vēl joprojām veic jau iepriekš zināmo klasifikācijas funkciju – neironu tīklā ir ievaddati, attēls ar datiem par pikseliem un no neironu tīkla tiek izvadīta klases vērtība, kas nosaka pie kādas klases pieder attēls vai tajā esošie objekti, tēli vai iezīmes. Tīkla arhitektūrā arī iekļauta zuduma funkcijas, piemēram, *SVM* vai *softmax* – rodas jautājums, ar ko tad konvolūciju neironu tīkli atšķiras no parastajiem mākslīgajiem neironu tīkliem?

Konvolūciju neironu tīklu uzbūves īpašība ir tā, ka tiek pieņemts, ka ievaddatu ieejā ir attēli, kas ļauj kopējā neironu tīklu arhitektūrā paredzēt un apstrādāt noteiktas, attēliem specifiskas, īpašības, kā piemēram – attēla izmēra datus, sarkano, zaļo un zilo krāsu kanālu pikseļu vērtības un citus ar attēlu saistītus datus. Nelielu attēlu gadījumā, piemēram, attēli no datu kopas CIFAR-10 ir ar izmēriem  $32 \times 32 \times 3$  (platums  $\times$  augstums  $\times$  krāsu kanālu skaits), pirmajam, pilnīgi savienotajam neironu slānim būtu  $32 * 32 * 3 = 3072$  svaru. Šāds skaits svaru varētu nešķīst pārāk liels, taču pilnīgi savienojamie neironu tīkla slāņi nav īsti piemēroti lielāku attēlu apstrādei, piemēram, attēlam ar izmēriem  $640 \times 480 \times 3$  (platums  $\times$  augstums  $\times$  krāsu kanālu skaits) pirmā, pilnīgi savienotā neironu tīkla slāņa neironiem būtu pat līdz  $640 * 480 * 3 = 921\,600$  svaru, kas jau ir ļoti liels un grūti glabājams un apstrādājams svaru skaits, tādēļ nav praktiski veidot neironu tīklu pēc klasiskās neironu tīklu uzbūves. [8]

Atšķirībā no parastajiem mākslīgajiem neironu tīkliem, konvolūciju neironu tīklu slāņos neironu ir organizēti trijās dimensijās – platuma, augstuma un dziļuma. Vērtīgi pieminēt, ka dziļuma dimensija neattiecas uz kopējo neirona tīkla dziļumu, bet gan uz neironu slāņu dziļumu.



2.1. att. Konvolūciju neironu tīklu uzbūves piemērs [8]

## 2.1. Konvolūciju neironu tīkla slāņi

Konvolūciju neironu tīklu uzbūves pamatā ir vairāki, nozīmīgi slāņi, kas tos atšķir no parastajiem neironu tīkliem, kas var tik iedalīti trīs galvenajos tipos:

1. Konvolūciju slānis (angļu: *convolutional layer*)
2. Kopošanas slānis (angļu: *pooling layer*)
3. Pilnās sasaistes slānis (angļu: *fully connected layer*)

Atsevišķos gadījumos, atkarībā no dažādām konvolūciju neironu tīklu arhitektūrām un uzbūves īpatnībām, tiek pielietoti arī citi transformācijas slāņi, kā piemēram, izmešanas slānis (angļu: *dropout layer*), normalizācijas slānis, savukārt kā aktivācijas funkcijas visbiežāk tiek lietotas jau iepriekšējā nodaļā minētās *ReLU* tipa veida funkcijas, kurām ir būtiskas veikspējas priekšrocības praktiskajā vidē. [8, 14]

Zemāk tiek aprakstīts piemērs konvolūciju neironu tīklu arhitektūrai priekš CIFAR-10 attēlu klasificēšanas. Šī uzdevuma veikšanai var tik izmantots konvolūciju neironu tīklu slāņu kārtojums sekojošā secībā – *Ieejas slānis, konvolūciju slānis, ReLU slānis, kopošanas slānis un pilnās sasaistes slānis*. [8]

Ieejas slānī tiek glabātas neapstrādātas ieejas attēlu pikseļu vērtības, kas dotajā piemēra gadījumā ir attēls ar platumu 32 pikseļi, augstumu 32 pikseļi un trīs – sarkano, zaļo, un zilo krāsu kanāliem.

Konvolūcijas slānī tiek aprēķinātas izejas vērtības, kas ir saistītas ar noteiktu, mazāku reģionu kopējā attēla kontekstā. Piemēram, ja tiek izmantoti 12 filtri, tad kopējais slāņa izmērs būtu  $32 \times 32 \times 12$ .

*ReLU* slānī tiek pielietota *ReLU* aktivācijas funkcija, kā piemēram  $\max(0, x)$ , katrai neironu izejas vērtībai, kas nemaina slāņa izmēru, t.i., tas paliek konvolūcijas slāņa izmērā –  $32 \times 32 \times 12$ .

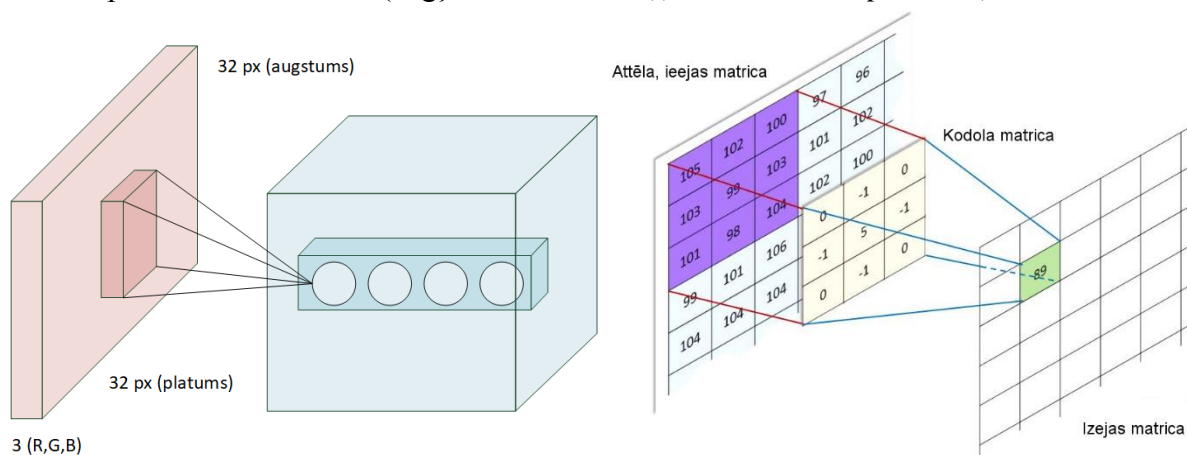
Kopošanas slānī notiek slāņa jeb iezīmju lieluma samazināšana attēla telpiskajiem izmēriem – platumam un augstumam kā rezultātā tiek iegūts slānis ar izmēru  $16 \times 16 \times 12$ .

Pilnās sasaistes slānī tiek aprēķinātas klasifikatoru gala vērtības attiecīgajām klasēm, kur piemērā tiek izmantotas 10 klases, kā rezultāta slāņa izmērs ir  $1 \times 1 \times 10$ , kur katrs no 10 skaitļiem atbilst noteiktai CIFAR-10 attēlu kopas attēlu kategorijai.

Šādā veidā konvolūcija neironu tīkls oriģinālo attēlu pikseļu vērtības, slāņi, pa slāņim pārveido par rezultātu, kas nosaka attēla kategorijas piederības. Svarīgi piebilst, ka ne visi slāņi satur parametrus, piemēram, konvolūciju un pilnās sasaistes slāņi veic transformācijas ne tikai no ieejas datiem, bet arī no neironu svaru un noslieces parametriem. [8]

## 2.1.1. Konvolūciju slānis

Konvolūciju slānis ir konvolūciju neironu tīklu pamats un galvenā sastāvdaļa, kas veicu grūtāko aprēķinu darbu salīdzinājumā ar pārējiem tīkla slāņiem. Konvolūciju slāņa parametri sastāv no apmācāmiem filtriem. Katrs filtrs pēc dimensijām (platuma un augstuma) ir mazāks, bet dziļumā tikpat dziļš kā ieejas dati. Piemēram, tipiski konvolūciju līmenis var sastāvēt no filtriem ar izmēriem 5x5x3 (piecu pikseļu platums un garums, trīs krāsu kanāli). Šādus filtrus dēvē arī par kodola matricām (angļu: *kernel matrix*), skatīt 2.2. att. (pa kreisi) [8,16]



2.2. att. Konvolūciju slāņa un kodola matricas uzbūve un funkcija [8, 17]

Šādas kodola matricas jeb filtra izmantošana ļauj konvolūciju slānim pēc slīdošā loga principa (angļu: *sliding window*) aprēķināt izejas vērtības balstoties uz filtru un ieejas datu skalāro reizinājumu summu. Šāda darbība ir nepieciešama, lai konvolūciju neironu tīkls šajā slānī spētu “apstaigāt” visus attēla reģionus, kas nedaudz pārklājās tādā veidā izvelkot un stimulējot neironus, kas spēj noteikt kādu objektu, tēlu vai iezīmi ievaddatos, skatīt 2.2 att. (pa labi), jo pilnīga un savstarpēja neironu savienošana starp slāņiem attēlu uzdevumos nav praktiska pielietojuma un tas šo uzdevumu skaitļošanas ziņā padara daudzkārt sarežģītāku.

Konvolūcijas slānim matemātiskās teorijas un modeļa pamatā ir matemātiska operācija – konvolūcija. Konvolūcija pēc būtības ir līdzīga savstarpējai korelācijai (angļu: *cross-correlation*) un tā ir aprakstāma ar formulu (2.1). [16]

$$s(t) = \int x(a)w(t - a)da \quad (2.1)$$

, kur funkcija  $x$  ir ieejas dati,  $w$  ir kodola matricas dati.

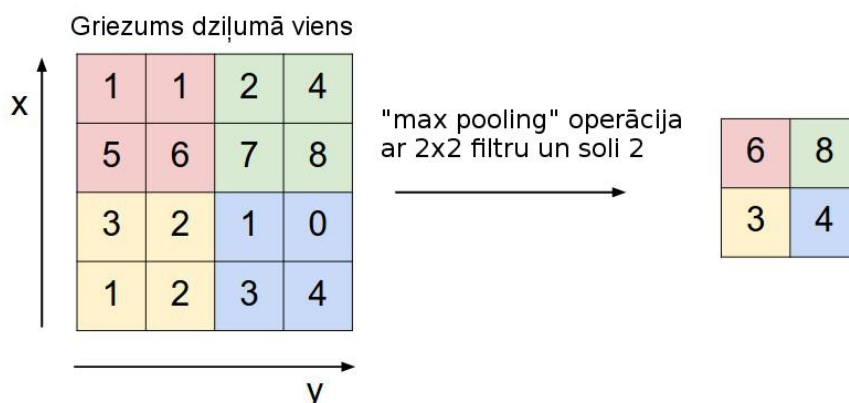
Vienkāršojot, konvolūciju neironu tīklā šāda operācija ļauj neironiem “ieraudzīt” noteiktus tēlus vai iezīmes, kur neironi pie attēla ieejas vērtību un trenētas kodola matricas, konvolūciju slāņa izejas vērtībās parāda tēlu kartes (angļu: *feature maps*). [8]

## 2.1.2. Kopošanas slānis

Viens no konvolūciju neironu tīklu tipiskajiem slāņiem ir kopošanas slānis, šie slāņi arī tiek dēvēti par regularizācijas slāņiem (angļu: *regularization layer*). Pēc pirmā konvolūciju slāņa, kur tiek izskaitļotas lineārās transformācijas seko otrais slānis, ko dēvē arī par detektora slāni, kurā tiek pielietotas nelineāras aktivitātes funkcijas (piem. *ReLU*). Tālāk seko kopošanas slānis, kas vēl vairāk modificē izejas datus no otrā slāņa.

Kopošanas funkcija aizvieto tīkla izejas vērtības ar summētajām izejas vektora blakus vērtībām, piemēram *max pooling* operācija atgriež maksimālo izejas vērtību noteiktā, taisnstūrveida tēlu apgabalā. Citas populāras kopošanas funkcijas izmanto nevis maksimālo, bet gan vidējo vērtību noteikšanu tēlu apgabalā.

Visos gadījumos kopošanas slānis palīdz izejas vērtībām un tēlam būt gandrīz invariantam attiecībā pret nelielām virzes kustībām ieejas datos. Tas nozīmē, ka ja ieejas dati tiek nedaudz virzīti, piemēram, tēls atrodas nedaudz citādākā, pagrieztā, leņķī vai mazliet citā vietā, tad kopošanas slāņa vērtības paliek nemainīgas. Šī ir svarīga īpašība, lai attēlos varētu noteikt tēlu un iezīmju esamību kopumā nevis tieši precīzi, kur šis tēls atrodas. [16]



2.3. att. Kopošanas slāņa un max pooling operācijas piemērs [8]

Visbiežāk tiek izmantots kopošanas slānis ar filtra izmēru 2x2 un soli 2, tas praktiski nozīmē to, ka no četriem blakus esošiem pikseliem tiek izvēlēts viens – ar vislielāko skaitlisko vērtību, skatīt 2.3. att. Kopošanas slāņa darbības rezultātā tiek arī samazināt apstrādājamās informācijas daudzums atkarībā no filtra un soļu izmēriem, taču attēla dziļuma informācija nemainās, tiek samazinātas tikai attēla platuma un augstuma dimensijas jeb telpiskais izmērs. Citas labi zināmas kopošanas slāņu funkcijas ir:

- *average pooling*
- *L2-norm pooling*

Pēdējo gadu laikā biežāk tiek izmantota *max pooling* operācija, jo prakse rāda kā tā strādā labāk un efektīvāk par augstāk minētajām funkcijām atkarībā no uzdevuma.[8, 16]

### 2.1.3. Pilnās sasaistes slānis

Konvolūciju neironu tīklos pilnās sasaistes slānis būtiski neatšķiras no jau iepriekšējā nodaļā apskatīto pilnās sasaistes slāni parastajos, mākslīgajos neironu tīklos. Šī slāņa darbības princips ir konvolūciju un kopošanas slāņos apstrādās informācijas grupēšana un tā kā šis slānis parasti ir pēdējais, tad izejas datus tiek izvadīts attēla apraksts, t.i., segmenti, klases, tēlu vai iezīmju atrašanās vietas vai cita informācija atkarībā no uzdevuma un izveidotā, konfigurētā un apmācītā neironu tīkla. Līdzīgi kā konvolūciju slānis, šis slānis ir pilnīgi, savstarpēji sasaistīts, taču atšķirība konvolūcija slānim ir tāda, ka tas ir telpiskā ziņā daudz mazāks un apskata tikai lokālu reģionu attēlā. Pilnās sasaistes slāni var uztvert kā konvolūcija slāni, kura redzes laukuma izmērs ir vienāds ar ieejas matricas izmēru. [8]

## 2.2. Konvolūciju neironu tīklu veidi

Konvolūciju neironu tīklu topoloģijas un arhitektūras attīstībā pēdējo gadu laikā ir noticis diezgan ievērojams progress, kas ir ļāvis šos neironu tīklus, ar attiecīgajām priekšrocībām un trūkumiem, piekoriņēt atbilstoši uzdevumiem, kā arī sasniegt ievērojamus ātrdarbības rezultātus un ļaut objektu un iezīmju noteikšanas uzdevumus attēlos vai video straumēs izpildīt reāllaikā. [18]

Sīkāk darbības principi, veiktspējas parametri, priekšrocības un trūkumi tiks apskatīti sekojošiem konvolūciju neironu tīklu veidiem:

1. *R-CNN* (angļu: *Region-based Convolutional Network*), kā arī to uzlabotās versijas *Fast R-CNN*, *Faster R-CNN* un *Mask R-CNN*;
2. *SSD* (angļu: *Single Shot Detector*), kā arī šai klasei līdzīgus modeļus, kā piemēram, *YOLO* (angļu: *You Only Look Once*);

Katrs no šiem veidiem ir ar savām niansēm un detaļām, kāpēc un kā katrs no tiem strādā un kāpēc viens ir labāks vai sliktāks par otru, nākamajās apakšnodaļās īsumā tiks apskatīti augstāk minēto konvolūciju neironu tīklu veidi balstoties uz publikācijām un citiem publiski pieejamiem informācijas avotiem, kur ir veikta šo tīklu darbības analīze un eksperimenti. Virspusējo arhitektūru kopsavilkumu var apskatīt darba beigās (skat. 1. pielikumu).

### 2.2.1. R-CNN

Reģionu bāzētais konvolūciju neironu tīkls pirmo reizi sabiedrības uzmanībā nonāca 2013. gada novembrī, kad tika izdots Rosa Giršika (angļu: *Ross Girshick*) un viņa līdzautoru tehniskās atskaites raksts par šo konvolūciju neironu tīkla uzbūves veidu, kas būtiski uzlabo

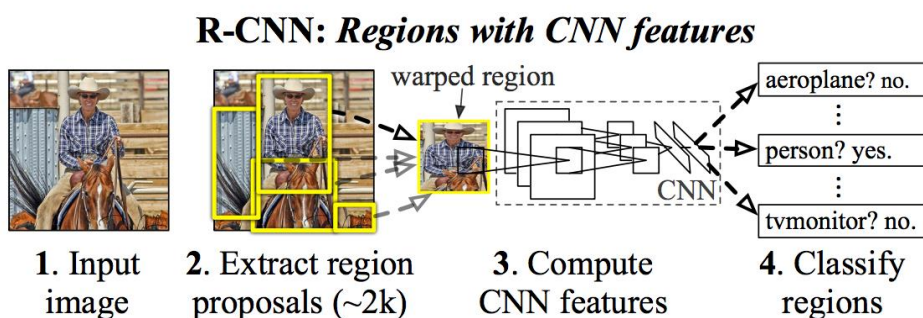
objektu un tēlu noteikšanu attēlos izmantojot viņu izstrādāto algoritmu. [18, 19]

*R-CNN* pamatā ir galveno objektu reģionu korekta noteikšana attēlā izmantojot robežtaisnstūri (angļu: *bounding box*). Ievaddatos tiek norādīts attēls un sagaidāmais rezultāts no šī konvolūciju neironu tīkla ir robežtaisnstūris un birka par katru atrasto objektu attēlā. Robežtaisnstūri tiek iegūti izmantojot selektīvo meklēšanu (angļu: *selective search*), skatīt 2.4 att., kas meklē līdzības attēlā caur dažāda izmēra logiem un blakus esošu pikseļu kopīgajām īpašībām šajos logos, kā piemēram, kopīga tekstūra, krāsa vai gaišums. [20, 29]



2.4. att. Selektīvās meklēšanas piemērs ar logu palīdzību [20]

Nākamais solis ir šo attēlā sameklēto robežtaisnstūru logu nodošana tālākai apstrādei. Tam tiek izmantots modificēts *AlexNet* [14] konvolūciju neironu tīkla modelis. Pēc šī pirmā soļa var būt ap 2000 robežtaisnstūru logu ieteikumu, kur katrs no logiem tiek padots konvolūciju neironu tīklam objektu klasificēšanai, skatīt 2.5. att. [18, 19]



2.5. att. R-CNN uzbūves shēma un apstrādes soļi [19]

Pēdējā solī tiek izmantota atbalsta vektora mašīna – *SVM* (angļu: *Support Vector Machine*), kas vienkāršojot – klasificē to vai tas vispār ir objekts un ja ir, tad kāds, kā tas ir redzams 2.5. att., ceturtajā solī.

Pats pēdējais solis, kas vēl vairāk uzlabot robežtaisnstūru lodziņu atbilstību meklējamajam objektam, izmanto lineārās regresijas (angļu: *linear regression*) modeli, kas tiek darbināts pēc klasifikācijas soļa – šādā veidā tiek panākts, ka izvadītie lodziņi precīzāk atbilst

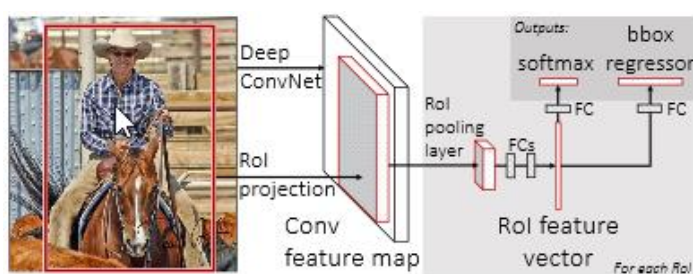
objekta izmēriem un raksturlielumiem. *R-CNN* darbojās pietiekami labi, pat par 7.1% ātrāk [19] izmantojot ILSVRC2013 izaicinājuma sacensību datu kopu salīdzinājumā ar otro labāko tīkla modeli *OverFeat* šo sacensību laikā, taču *R-CNN* arhitektūrai ir arī savi trūkumi:

1. Katrs sākotnēji sameklētais robežtaisnstūra logs, katram attēlam ir jāpadod konvolūciju neironu tīklam, lai veiktu klasificēšanu, kas 2000 lodziņu gadījumā būtu 2000 iterācijas priekš viena attēla.
2. Šīs arhitektūras darbībai ir nepieciešams apmācīt trīs atsevišķus modeļus – konvolūciju neironu tīklu, lai ģenerētu attēla iezīmes, klasifikatoru, kas paredz iezīmju klases, un regresijas modeli, kas sašaurina oriģinālo robežtaisnstūru logus, lai tie labāk atbilstu objektam.

Šo iemeslu dēļ šāda arhitektūra ir salīdzinoši grūti apmācām ir vairāki punkti, kur var tikt pieļautas kļūdas un nepilnības, kas ietekmētus algoritma darbību un precizitāti. [18, 19]

### 2.2.2. Fast R-CNN

Ross Giršiks, 2015. gadā atrisināja abus augstāk minētos *R-CNN* arhitektūras trūkumus, ar jaunu arhitektūru *Fast R-CNN*, kur vairs nebija nepieciešams uz vienu attēlu darbināt 2000 iterācijas ar klasifikatoru neironu tīklu – tā vietā to vajadzēja izdarīt tikai vienu reizi uz katru attēlu. Robežtaisnstūru logu noteikšana tika iekļauta klasifikatora konvolūciju neironu tīklā, kur tiek izmantots interesējošo, t.s., *RoI* (angļu: *Region of Interest*) reģionu kopošanas slānis – *RoIPool*. Šajā gadījumā *CNN* iezīmju kartes slānis tiek kopīgi izmantots priekš *RoIPool*, kur katra no atrastajām iezīmēm tiek saglabāta kopošanas slānī. Arī regresijas modelis tika iekļauts vienā kopējā neironu tīklā, skatīt 2.6. att. [22]



2.6. att. **Fast R-CNN** uzbūves shēma [22]

Atšķirībā no *R-CNN*, šādā arhitektūras modelī ir nepieciešams apmācīt tikai vienu neironu tīklu, lai izpildītu objektu atpazīšanas uzdevumu, bez tam tas arī ir ātrāk apmācāms un testējams un galvenais – ar augstāku vidējo klasificēšanas precizitāti salīdzinājumā ar *R-CNN* arhitektūru [18, 22]. Vēl viena būtiska īpašība, ka šajā arhitektūrā *AlexNet* konvolūciju neironu tīkla tiek izmantota *VGG-16* konvolūciju neironu tīkla modelis, kuru izstrādāja pētnieku grupa Oksfordas Universitātē (angļu: *University of Oxford*), arī SVM klasifikators tika aizvietots ar *softmax*

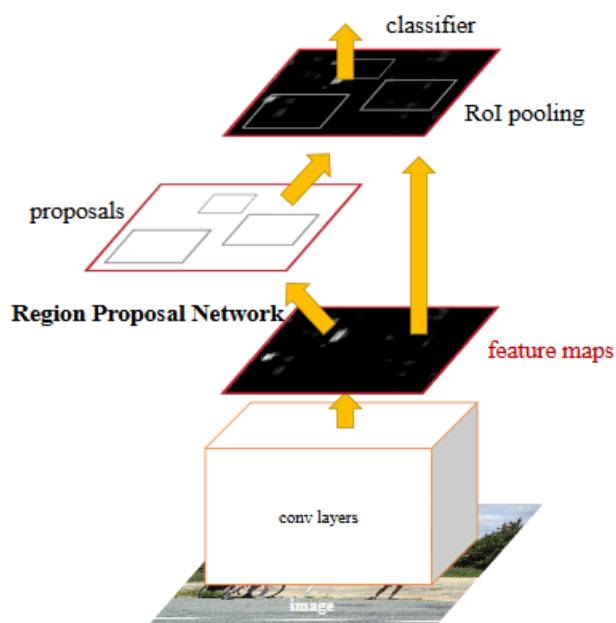
slāni, rezultātā ieejā ir attēli ar reģionu ieteikumiem, izejā – klasificēti objekti ar ciešāku robežtaisnstūra laukumu klasificētajam objektam. Atsevišķi *Fast R-CNN* parametru un hiperparametru konfigurēšanai, ir iespējama arī *VGG-16* parametru konfigurācija, lai šo tīkla modeli varētu pielāgot konkrētām vajadzībām. [18, 22, 23]

### 2.2.3. Faster R-CNN

Neskatoties uz līdzšinējiem uzlabojumiem *R-CNN* modeļos, tajā bija vēl viens trūkums – attēla reģionu ieteicējs (angļu: *region proposer*). Kā jau iepriekš tika aprakstīts, tad pirmais solis objektu noteikšanai *Fast R-CNN* arhitektūrā ir, pirmkārt, robežtaisnstūru logu ģenerēšana vai interesējošo reģionu testēšana. Līdzšinējos modeļos šo soļu izpildei tika izmantota selektīvā meklēšana, [20] kas ir salīdzinoši lēns un darbietilpīgs process, kas negatīvi ietekmēja kopējo modeļa ātrdarbību. [18, 24]

2015. gada vidū, pētnieku komanda no Microsoft Research, kuru vidū arī bija Ross Giršiks, atrada jaunu veidu, kā reģionu ieteikšanas soli varētu iekļaut jaunā tīkla arhitektūrā, kas praktiski nozīmētu to, ka šim solim vispār nevajadzētu izmantot, skaitļošanas ziņā salīdzinoši dārgo, selektīvās meklēšanas algoritmu.

Risinājums bija t.s. reģionu ieteikšanas tīkls jeb *RPN* (angļu: *Region Proposal Network*), kura pamatā bija jau esošo *CNN* slāņu izmantošana, lai gan veiktu attēla reģionu ieteikšanu, gan objektu klasificēšanu, skatīt 2.7. att.

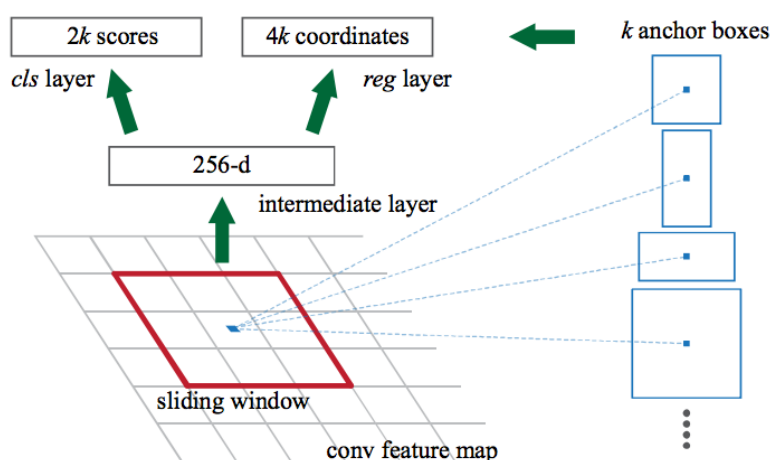


2.7. att. **Faster R-CNN** uzbūves shēma [24]

Pētnieki novēroja, ka konvolūciju iezīmju kartes, kas tiek izmantotas reģionu bāzētajos noteicējos, kā *Fast R-CNN*, var arī tikt izmantotas iesakāmo reģionu ģenerēšanā.

Pamata modelim ieejā tiek padoti attēli un izejā tiek sagaidīti klasificēti objekti ar robežtaisnstūru logu koordinātēm. [18, 24]

*RPN* jeb reģionu ieteikšanas tīkla darbību var raksturot šādi – vispirms, ar slīdoša loga metodi tiek iteratīvi apstaigātas konvolūciju neironu tīkla iezīmju kartes, kur pie katra loga tiek izvadīts  $k$  potenciālo robežtaisnstūri un rezultāts ar to, cik labi šis lodziņš atbilst objektam. Šie  $k$  robežtaisnstūri tiek dēvēti arī par  $k$  enkuru taisnstūriem (angļu: *k anchor boxes*). Šajos elementos tiek uzglabāti taisnstūru veidi un to proporcijas atkarībā no objektiem, piemēram, cilvēkus vislabāk raksturo vertikāli garsais taisnstūris un ir maz ticams, ka ļoti tievs vai citādākas formas taisnstūris atbilstu labāk cilvēku formas objekta raksturošanai, skatīt 2.8. att.



2.8. att. **RPN uzbūves shēma** [24]

*RPN* modeļa ievadā tiek padota *CNN* iezīmju karte un izejā tiek sagaidīts robežtaisnstūris katram enkuram un rezultāts ar to cik labi attēls šajā robežtaisnstūrī atbilst objektam. Tālāk izvads no *RPN* tiek padots uz *Fast R-CNN* modeli, kur notiek objektu klasifikācija un robežtaisnstūru pielāgošana.

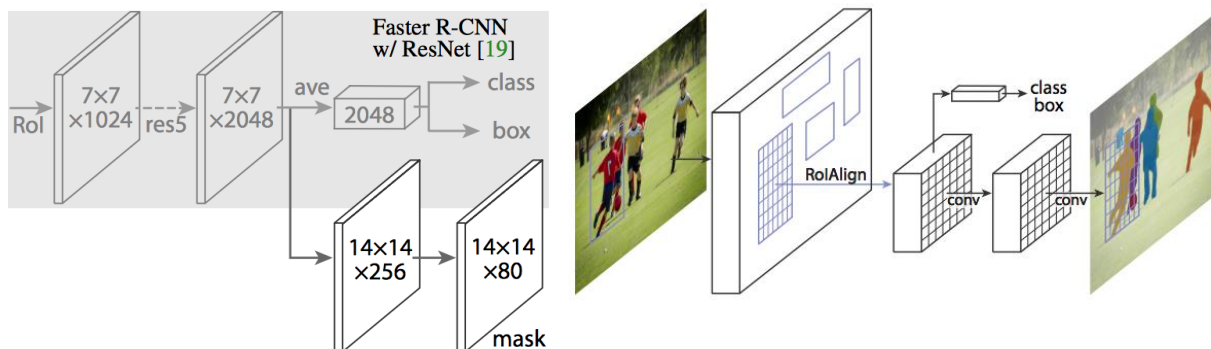
Šādu uzlabojumu veikšana rezultātā ļāva kārtējo reizi vienkāršot un uzlabot *R-CNN* modeļa apmācības ātrumu, testēšanu un ātrdarbību. [18, 24, 29]

## 2.2.4. Mask R-CNN

Viens no jaunākajiem *R-CNN* modeļiem ir 2017. gadā publicētais Facebook AI pētnieka Kaiminga Hī (angļu: *Kaiming He*) un viņa komandas izstrādātais *Mask R-CNN* modelis. [25] Šis modelis ir par attīstības soli tālāk, jo šeit tiek noteikta ne tikai objektu piederība konkrētai klasei un robežtaisnstūra logs, kur atrodas objekts, bet arī tiek iekrāsots jeb segmentēts objekts pikseļu līmenī.

*Mask R-CNN* modelī tas tiek panākts ar jauna zara pievienošanu esošajam *Faster R-CNN*

modelim, skatīt 2.9. att. (*pa kreisi*), kas atgriež bināru masku (angļu: *binary mask*), kas pasaka vai konkrētais pikselis ir vai nav daļa no objekta. No *CNN* iezīmju kartes tiek ģenerēta matrica jeb binārā maska ar vieniniekiem, kur objekts atbilst objektam, un nullēm, kur neatbilst.



2.9. att. **Mask R-CNN uzbūves shēma** [18, 25]

Lai pilnvērtīgi izmantotu binārās maskas priekšrocības, bija nepieciešams pamainīt *RoIPool* modeli, kas tika pārsaukts par *RoIAlign*. Šī nelielā izmaiņa bija nepieciešama, jo sākotnējais *Faster R-CNN* arhitektūras modelis ar *RoIPool* segmentēto objektu daļu atgriezta ar nelielu nobīdi no oriģinālā attēla, skatīt 2.9. att. (*pa labi*).

Šī problēma tika atrisināta izņemot noapaļošanu, kas tika veikta *RoIPOOL* slānī un to aizstājot ar bilineāro interpolāciju, šādā veidā objektu segmentācija pikseļu līmenī kļuva daudz precīzāka. [18, 25]

## 2.2.5. Single Shot Detectors

*SSD* (angļu: *Single Shot Detector*) tīklu arhitektūra, skat. 2.11. att., ir izstrādāta objektu noteikšana attēlos reāllaikā, t.i., objektu noteikšanai un klasificēšanai un noteikšanai tīkla ievadā varētu padot video straumi un šis tīkls varētu arī reāllaikā šos datus apstrādāt un atgriezt video straumi ar atrastajiem un klasificētajiem objektiem. Salīdzinājumā ar *Faster R-CNN*, šāds tīkls pie zemākas izšķirtspējas ievaddatiem spēja apstrādāt līdz pat 46 kadriem sekundē, kamēr reģionu bāzētais *Faster R-CNN* tīkls pie līdzīgas izšķirtspējas ievaddatiem spēj apstrādāt tikai līdz septiņiem kadriem sekundē. [28]

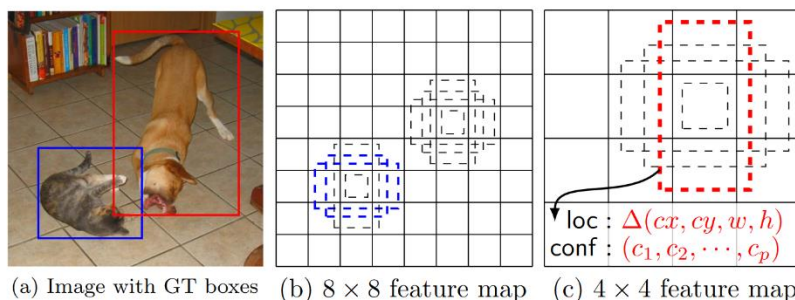
*SSD* tīkla arhitektūrā objektu, tēlu un iezīmju noteikšana attēlā sastāv no divām daļām:

1. Iezīmju karšu izvilkšana (angļu: *feature map extraction*);
2. Konvolūciju filtru pielietošana objektu noteikšanai;

Šajā tīkla arhitektūrā netiek izmantots atsevišķs tīkls, lai iegūtu attēla reģionu ieteikumus objektu noteikšanai, tā vietā tiek izmantota ļoti vienkārša metode – klašu rezultātu un objektu atrašanās vieta tiek aprēķināta nelielajos konvolūcija neironu tīkla slāņos. Pēc iezīmju karšu

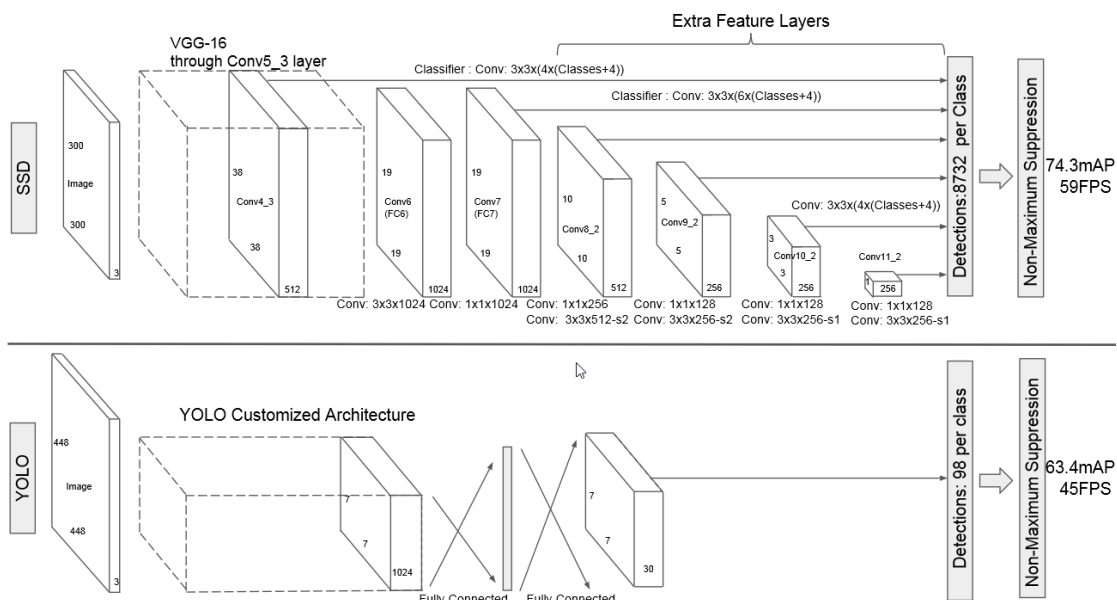
iegūšanas, *SSD* tīkla arhitektūrā tipiski tiek pielietoti 3x3 konvolūciju filtri, kas prognozē iezīmes piederību klasei, kā arī tās robežtaisnstūri – līdzīgi kā *RPN* modelī, kas tiek atsevišķi izmantots objektu noteikšanai *R-CNN* tipa tīklos.

Vēl viena svarīga *SSD* tīklu īpašība ir, ka atsevišķos konvolūciju neironu līmeņos notiek dažādu izmēru iezīmju karšu ģenerēšana no kurām attiecīgi var noteikt vai nu lielāka izmēra objektus vai mazāka izmēra objektus, skatīt. 2.10. att., kur ir redzams, ka mazākas izšķirtspējas iezīmju karte spēj noteikt lielāka izmēra objektus, kā piemēram – suni, taču lielākas izšķirtspējas iezīmju karte spēj noteikt mazāka izmēra objektus, kā piemēram – kaķi. [26, 28]



2.10. att. *SSD* iezīmju karšu izmantošana objektu noteikšanā [26]

Pie šīs pašas klases var arī pieskaitīt vēl vienu salīdzinoši jaunu un populāru *SSD* tipa tīklu – *YOLO* (angļu: *You Only Look Once*). Šī tīkla pamatā ir *DarkNet* neironu tīklu ietvaru, kas atbild par iezīmju karšu veidošanu un tīklā pēdējos slāņos atrodas konvolūciju neironu slāņi, kas nosaka objektu robežtaisnstūrus un klases. [27, 30]



2.11. att. *SSD* un *YOLO* tīklu uzbūves shēma [26]

Šo tīklu sliktā īpašība ir tā, ka tiem ir samazināta precizitāte mazu objektu noteikšanā, kam ir būtiska loma ceļa seguma iezīmju noteikšanā, jo ceļa seguma defekti būt nelieli.

## 2.3. Secinājumi

Pēc konvolūciju neironu tīklu veidu apskatīšanas un vairāku zinātniski pētniecisko un citu materiālu izpētes šajā nodaļā tiks aprakstītas galvenās gūtās atziņas un secinājumi.

Konvolūciju neironu tīkli ir visnotaļ piemēroti izvēlētajās problēmas risināšanā, to pierāda šo tīklu veiksmīgā izmantošana dažādu attēlu datu kopu apzināšanā ar diezgan augstiem precizitātes un veiktspējas rādītājiem. Pēdējie seši gadi objektu, tēlu un iezīmju atpazīšanā attēlos ar konvolūciju neironu tīklu palīdzību ir bijuši sevišķi nozīmīgi, jo šajā laikā ir veikti daudzi nozīmīgi atklājumi un pētniecības darbi [14, 19-28] šīs nozares attīstībā, kas ir ļāvuši, kopā ar tehnoloģisko attīstību, konvolūciju neironu tīkliem nonākt plašākas sabiedrības uzmanības centrā.

Konvolūciju neironu tīklu veidiem ir savas priekšrocības un trūkumi. Vairākos informācijas avotos [28-30] un pētījumos [19-27, 31] bija veikta šo konvolūciju neironu tīklu veidu veiktspējas un precizitātes (attiecībā uz dažādām attēlu kopām) testēšana dažādās tīklu arhitektūrās un vidēs, kā arī tika veikta konfigurācijas, parametru, hiperparametru u.c. maināmo lielumu testēšana, piemēram, par izšķirtspēju ietekmi uz veiktspēju (skat. 2. pielikumu).

Galvenās atziņas no šiem avotiem attiecībā uz konvolūciju neironu tīklu veida izvēli ir:

- *SSD* veida neironu tīkli ir labāk piemēroti reāllaika uzdevumiem, kur ir svarīga ātrdarbība un noteikšanas un klasifikācijas precizitāte ir sekundāra. Piemēram, šādu tīklu izmantošanai ir potenciāls autonomajās automašīnās.
- *SSD* darbojas sliktāk par *R-CNN* tipa tīkliem mazizmēra objektu atpazīšanā. Mazizmēra objekti var tikt atpazīti tikai augstas izšķirtspējas līmeņos, kas ir mazāk derīgi klasifikācijai.
- *SSD* salīdzinājumā ar *R-CNN* ir mazāka lokalizācijas kļūda, taču lielāka klasifikācijas kļūda, kad jāatpazīst līdzīgu kategoriju objekti.

Kā redzams, tad ceļu seguma iezīmju noteikšanai attēlos būtu labāk piemērots *R-CNN* konvolūciju neironu tīklu veids un attiecīgās arhitektūras, savukārt, ja ceļā seguma iezīmju noteikšanas uzdevums būtu jāveic reāllaikā, tad šim uzdevumam būtu labāk piemēroti *SSD* tipa neironu tīkli.

Vērtīgi atcerēties, ka neirona tīkla kopējo darbību (apmācīšanos, testēšanu, atpazīšanu) un veiktspēju ietekmē arī tīkla un izmantojamā mašīnmācīšanās ietvara pielāgošanai konkrētajam uzdevumam.

Būtiska loma uzdevuma veiksmīgā izpildē ir arī plašas, anotētas, attēlu kopas sagatavošanai, kas tiktu izmantota neironu tīkla apmācīšanas procesā.

### 3. DZIĻĀS MAŠĪNMĀCĪŠANĀS IETVARI

Neironu tīklu izveidošana, konfigurēšana un apmācīšana ir kļuvusi vieglāka pateicoties mūsdienās pieejamajiem mašīnmācīšanās ietvāriem. Šajā sadaļā tiek apskatīti daži populārākie un pētnieciskajos avotos pieminētie mašīnmācīšanās ietvari – TensorFlow, Caffe un PyTorch. Šie ietvari ir izvēlēti pamatojoties uz faktu, ka šie ietvari tika izmantoti dažādu konvolūciju neironu tīklu izveidē [19, 22, 24, 25] un tika uzrādīti kopā ar publicēto tīklu veidu pirmkodu kā ieteicamie ietvari. Mūsdienās eksistē daudz mašīnmācīšanās ietvaru (skat. 3. pielikumu), no kuriem liels skaits ir saistīti ar tādiem lielajiem tehnoloģiju uzņēmumiem kā: *Google*, *Facebook*, *Amazon*, un *Microsoft*.

#### 3.1. TensorFlow

*TensorFlow* ir viens no vispopulārākajiem [34] mašīnmācīšanās ietvāriem, kas tiek izmantots mūsdienās. *TensorFlow* izstrādā *Google*, kas mērķi aizvīdot *Theano* mašīnmācīšanās ietvaru kaut arī patiesībā šie ietvari ir diezgan līdzīgi. Daži no *Theano* izstrādātājiem, kā piemēram Īans Gudfelovs (angļu: *Ian Goodfellow*) palīdzēja izveidot *TensorFlow* pirms pameta *Google*, lai pievienotos *OpenAI*.

*TensorFlow* ir dažas problēmas ar *inline* matricu operācijām, kuras spiestā kārtā ir jāpilda ar matricu kopēšanu un lielām matricām šī operācija pati par sevi dārga no skaitļošanas viedokļa. Dažas darbības ar *TensorFlow* var aizņemt līdz par četrām reizēm ilgāk kā citi augstākas klases mašīnmācīšanās ietvari, kā piemēram, *MxNet* un *CNTK*, taču *Google* strādā pie šo problēmu atrisināšanas.

*TensorFlow* ir uz dziļo mašīnmācīšanos orientēts ietvars, taču tam ir arī atbalsts pastiprinātajai apmācīšanai un citiem algoritmiem.

*TensorFlow* priekšrocības: atbalsta *Python* un *NumPy*, skaitļošanas grafu abstrakcija, ātrāka kompilēšana kā *Theano*, *TensorBoard* priekš vizualizācijas, ļoti labi atbalsta skaitļošanas mērogošanu (angļu: *computing scaling*), kas krietni uzlabo tā ātrdarbību.

*TensorFlow* trūkumi: nedaudz lēnāks kā citi augstas klases ietvari, daudz vairāk piesātināts ar dažādām iespējām, kas to dažkārt padara grūtāk lietojamu, nav apmācītu modeļu piemēru, nav komerciāla nolūka atbalsts, grūtāk integrējams ar citiem rīkiem.

Ietvarā esošais *API*, jeb lietojumprogrammas interfeiss atbalsta sekojošās programmēšanas valodas: *Python*, *C++*, *Java*, *Go*, *Swift*. Pamatā galvenais atbalsts tiek nodrošināts *Python* saskarnei. [32, 33]

### 3.2. Caffe un Caffe2

*Caffe* ir labi pazīstams un plaši pielietots datorredzes uzdevumos. *Caffe* izmantoja ātro konvolūciju neironu tīklu *Matlab* implementāciju, kura pēc tam tika pārnesta uz *C* un *C++*.

*Caffe* nav paredzēts tādiem dziļās mašīnmācīšanās uzdevumiem, kas satur darbības ar tekstu, skaņu vai laika datu sērijām. Līdzīgi kā citi mašīnmācīšanās ietvari arī šī ietvara *API* pamatā ir *Python* programmēšanas valoda.

*Caffe* priekšrocības: Labi pielāgots vienvirziena neironu tīkliem un attēlu apstrādes uzdevumiem, labi darbojas ar esošo neironu tīklu pielāgošanas ziņā, ir iespējams apmācīt modeļus nerakstot kodu, *Python* lietojumprogrammas interfeiss ir labi izstrādāts un lietderīgs

*Caffe* trūkumi: Neatbalsta *CUDA* lietojumprogrammas interfeisu un jaunāko videokaršu procesorus kā rezultātā ir jāveic pašrocīga pielāgošana rakstot slāņus *C++* valodā, nav labi piemērots atkārtoto neironu tīkliem, grūti strādā ar lieliem neironu tīklu modeļiem kā piemēram *GoogLeNet* un *ResNet*, nav iespējama ietvara paplašināšana, nav komerciāla nolūka atbalsts, aizstāts ar *Caffe2*

Aktīva izstrāde pie *Caffe* vairs nenotiek, pēdējā stabilā versija tika izlaista 2017. gada aprīlī. [32,35]

2017. gadā plašākai publikai tika izlaista *Caffe2* ietvara versija, kas ietvēra vairākus uzlabojumus no *Caffe* ietvara – pirmās klases atbalsts plaša mēroga izklieģētai apmācīšanai (angļu: *distributed training*), mobila izvietošana, jaunākās datortehnikas atbalsts, tajā skaitā jaunāko *CPU* un *CUDA* atbalsts. [32, 36]

### 3.3. Torch un PyTorch

*Torch* ir skaitļošanas ietvars ar *API*, kas ir rakstīts *Lua* programmēšanas valodā un atbalsta mašīnmācīšanās algoritmus. Dažas *Torch* versijas tiek izmantotas lielo tehnoloģiju kompāniju *Facebook* un *Twitter* iekšējo dziļo mašīnmācīšanos platformu pielāgošanai.

*Torch* ietvarā neironu tīklu definēšanā tiek izmantots slāņu grafs, kas ļauj ātri un bez apgrūtinājumiem izveidot neironu tīklu ar jau eksistējošiem slāņiem, taču raupjas slāņu grafa detalizācijas pakāpes dēļ ir sarežģītāk ir jaunu slāņu tipus.

Jauno slāņu veidošana ir salīdzinoši vieglāka kā tas ir *Caffe* ietvarā, kur tiek izmantota *C++* valoda. *Torch* ietvars tiek uzskatīts par ļoti labu variantu konvolūciju neironu tīklu implementēšanai. [39]

*Torch* ietvars balstās uz *Lua* programmēšanas valodu un *LuaJIT* kompilatoru. *Torch* ir labi optimizēts un ātrdarbīgs ietvars salīdzinājumā, piemēram, ar *TensorFlow*. [32, 39]

Viens no šī ietvar mīnusiem ir tas, ka *Lua* programmēšanas valoda nav pārāk izplatīta un tas var radīt grūtības ar šī ietvara integrēšanu citos rīkos.

*Torch* paralēli arī tiek izstrādāts ietvars *PyTorch*, kas ir balstīts ir *Python* programmēšanas valodu, kas paver plašākas *API* izmantošanas un integrācijas iespējas ar citiem rīkiem, tajā pat laikā *Python* ir daudzkārt populārāka valoda par *Lua*, kas daudziem cilvēkiem ir kā priekšrocība.

Starp *Torch* un *PyTorch* pastāv nelielas atšķirības, tomēr šie ietvari pēc būtības ir visai līdzīgi.

*Torch* un *PyTorch* priekšrocības: daudz modulāru daļu, kuras var viegli kombinēt, viegli izveidot jaunu slāņu tipus un veikt skaitļošanu izmantoju video kartes procesoru, liels skaits ar iepriekš apmācītiem modeļiem.

*Torch* un *PyTorch* trūkumi: parasti ir nepieciešams rakstīt savu apmācības kodu, mazāk iespēju izmantot gatavu piemēru, nav komerciāla nolūka atbalsts, salīdzinoši zemas kvalitātes dokumentācija. [32, 37, 38]

### 3.4. Secinājumi

Mūsdienās ir pieejams liels skaits dziļas māšīnmācīšanās ietvaru, katri ar savām priekšrocībām un trūkumiem, taču apskatītajos informācijas avotos [32, 34, 39], diezgan bieži pozitīvu novērtējumu saņem tieši *PyTorch*, *Caffe2*, *TensorFlow* un *Keras* māšīnmācīšanās ietvari.

*Keras* ir augstāka līmeņa ietvars, kas izmanto *TensorFlow* un *Theano* ietvarus. Šis ietvars un *Python* lietotjumprogrammas interfeiss arī ir ļoti augstu novērtēts – kā viens no visintuitīvākajiem un vieglāk lietojamajiem interfeisiem, kas ir ieviests no *Torch* un *PyTorch* ietvariem.

Gandrīz visi iepriekš minētie, kā arī citi dziļās māšīnmācīšanās ietvari nepārtraukti tiek izstrādāti un uzlaboti – tiem parādās jaunas iespējas un funkcionalitāte, kā arī tiek uzlabota veiktspēja.

Galvenās atziņas, kas tika iegūtas veicot virspusēju dziļās māšīnmācīšanās ietvaru apskatu ir:

- Ir nepieciešams izvēlēties atbilstošo ietvaru konkrētā uzdevuma veikšanai, piemēram, daži ietvari neatbalsta citu neironu tīklus, kā tikai tāds, kas paredzēti attēlu apstrādei.
- Jaunāki un aktīvi izstrādāti neironu tīkli būs vieglāk lietojami un ar plašāku funkcionalitāti, kā arī lietotāju kopienas atbalstu.

## 4. CEĻA SEGUMA IEZĪMJU NOTEIKŠANAS PROBLĒMAS ANALĪZE

Šajā nodaļā tiks analizēta ceļu seguma virsmas īpatnības un kā tās varētu ietekmēt dziļās mašīnmācīšanās pielietošanu ceļa segumu iezīmju noteikšanā, papildus tam tiks apskatīti citi veiktie pētījumi šajā jomā un uzskaitīti kādi alternatīvie risinājuma veidi vēl pastāv šīs problēmas risināšanai.

Ceļa seguma bojājumi nav tikai problēma ar Latvijas ceļiem, bet gan visā pasaulē. Bojātie ceļu posmi nereti ir pilnībā jāpārbūvē, ja bojājumi nav tikuši novērsti laikus, kas rada lielas izmaksas ceļu uzturētājiem. Balstoties uz pētījumā [40] veikto analīzi var veikt noteiktus pieņēmumus par potenciālajām problēmām praksē.

### 4.1. Problēmas analīze

Lai laikus varētu veikt ceļa seguma virskārtas atjaunošanas darbus, ir nepieciešams veikt seguma stāvokļa novērtēšanu. Kā viena no metrikām šim mērķim tiek izmantots seguma stāvokļa indekss - *PCI*.

*PCI* ir salīdzinoši vienkāršs veids, kā tiek veikta ceļa segas virskārtas novērtēšana – šī metode ļauj identificēt nepieciešamos uzturēšanas un atjaunošanas darbus visā valsts ceļu tīklā šādi nodrošinot to, ka ceļu uzturēšanai paredzētie finansiālie un materiālie līdzekļi tiek izlietoti lietderīgi. [40]

*PCI* indekss apraksta divus kritērijus:

- seguma virsmas bojājumu veidu, apmēru, pakāpi (parasti plaisas, rīses);
- līdzenumu un braukšanas komfortu;

*PCI* pēc būtības ir subjektīva ceļu seguma stāvokļa novērtēšanas metode, kas balstīta uz seguma stāvokļa inspekcijām, stāvokļa izmaiņu novērošanu un sistemātisku novērtēšanu, ko veic pieredzējuši speciālisti apbraukājot ceļu tīklus. Seguma stāvokļa indeksa aprēķināšanas veikšana ir vēlama katru gadu, lai varētu novērtēt ceļu segas pasliktināšanās tempus.

Kā noprotams tad *PCI* ir būtiska loma ceļu uzturēšanas procesā, kas varētu tikt uzlabots ar mašīnmācīšanās palīdzību, piemēram, šādas sistēmas izstrādes rezultātā ceļu speciālistiem potenciāli varētu tikt atvieglots darbs *PCI* aprēķināšanā – ceļa stāvoklis un bojājumu veidi, apmērs un to atrašanās vieta uz ceļa varētu tikt noteikta ar konvolūciju neironu tīklu palīdzību un automašīnā uzstādītu video vai fotokameru. Atkarība no prasībām bojājumu klasifikācija ar zināmiem trūkumiem var tikt veikta reāllaikā, t.i. braucošā mašīnā. Alternatīvs variants būtu *GPS* uztvērēja izmantošana un attēlu vai video fragmentu pēcapstrāde ar neironu tīklu, kas tiktu korelēta ar *GPS* datiem, lai noteiktu bojāto ceļu posmu atrašanās vietu.

Šādu sistēmu potenciāli varētu apmācīt atpazīt dažādu ceļa segumu tipu bojājumu veidus, piemēram, grants vai bruģa ceļa seguma bojājuma veidi, lielākoties, ir vizuāli atšķirīgi no bituminēta jeb melnā ceļa seguma bojājuma veidiem.

Autoceļa seguma novērtēšanai virsmas bojājumus iedala trīs grupās:

1. Seguma virsmas defekti
  - a. Defragmentācija (bedres);
  - b. Seguma “svīšana” (izsvīdumi);
  - c. Nodilums;
2. Seguma virsmas deformācijas
  - a. Seguma viļņošanās un stumšanās;
  - b. Rises;
3. Plaisas
  - a. Garenplaisas;
  - b. Nogruma;
  - c. Garenšuvu plaisāšana;
  - d. Seguma malu plaisāšana;
  - e. Šķērsplaisas;

Šo defektu grupām praktiskos nolūkos ir savi svāri, kas nosaka to, cik nopietns ir ceļa seguma bojājums (skat. 4. pielikumu).

## 4.2. Attēlu iezīmju klases

Konvolūciju neironu tīklu izteiksmē ir ļoti svarīgi saprast, kādas iezīmes vai objekti attēlā ir jānosaka, no tā ir atkarīga neirona precizitāte uzdevuma veikšanā un neironu tīkla apmācības kopas sagatavošanā. Pētījuma [40] aprakstītajām seguma virsmas bojājumu klasēm ir pieejami iezīmju attēlu piemēri, kas tiks attēloti un aprakstīti šajā nodaļā.

Pirmā iezīmju, kategorija jeb klase ir plaisas. Plaisu kategorijā ir izteiktas daudzas, un nereti neliela izmēra iezīmes, skat. 4.1. att. Šādai iezīmju kategorijai ir potenciāli daudz dažādu kombināciju un variantu iezīmju attēlojumam, kas var diezgan būtiski atšķirties vienas klases ietvaros, taču tās visas vieno viena pazīme – attēlos ir daudz līniju, kur blakus esošo pikseļu vērtības stipri atšķiras, t.i., salīdzinoši gaišais ceļa segums pretstatā tumšajā plaisām. Viens no problemātiskajiem gadījumiem varētu būt laboto plaisu skat 4.1. att (f) atšķiršana no nelabotajām plaisām, skat. 4.1. att (a, b). Ir redzams, ka laboto plaisu gadījumā, tumšā līnijas ir daudz biezākas, taču perspektīvas rezultātā šādas labotās plaisa, ka attēlā atrodas tālāk no kameras varētu tikt kļūdaini klasificētas, kā plaisas.



(a) Noguruma plaisas



(b) Blokveida plaisas



(c) Seguma nomales plaisas



(d) Seguma garenplaisas



(e) Seguma šķērsplaisa (termoplaisa)



(f) Apstrādātas blokveida plaisas

#### 4.1. att. Plaisu iezīmes [40]

Otrā iezīmju klase ir seguma virsmas defekti. Bedrīšu noteikšana ar datorredzi un mašīnmācīšanos jau ir tikusi demonstrēta pētniecības darbos [41, 42], kur tika izmantoti *LBP* (angļu: *Local Binary Patterns*) un *SVM + HOG* (angļu: *Histogram of oriented gradients*) klasifikatori bedrīšu noteikšanai. Svarīgi atzīmēt, ka bedres ir dažādas formas un izskata, tās savu tekstūra attēlā var mainīt atkarībā no dienas stundas, ēnu krituma vai laikapstākļiem. Otrs faktors, ko vajadzētu ņemt vērā, ka dažas bedres var būt regulāras formas, piemēram, frēzēts

remontdarbu laukums uz ceļa seguma, tāpat pēc šo regulāro bedru aiztaisīšanas, ielāpi parasti ir stipri atšķirīgi no pārējā ceļa seguma, taču tas gluži vairs nav ceļa seguma bojājums, skatīt. 4.2. att.



(a) Bedre



(b) Bedres ielāps

#### 4.2. att. **Bedrīšu un ielāpu iezīmes** [40]

Pie seguma virsmas defektiem ietilpst arī virsmas “svīšana”, skat. 4.3. att., kas izpaužas ar bituminētā slāņa izspiešanos uz ceļa seguma virsmas, šāda iezīme līdzinās uz mitra ceļa atstātu riepu nospiedumam, tikai pretēji. Attēlā redzamā iezīme var būt stipri izteikta – realitātē ir iespējams, ka uz ceļa tikai atsevišķos posmos vai skatiena leņķos ir pamanāms šis seguma virsmas defekts.



#### 4.3. att. **Seguma “svīšana”** [40]

Visbeidzot, pie seguma virsmas defektiem tiek ieskaitītas arī ceļa nodiluma un izdrupumu iezīmes. Ceļa nodilumu varētu vislabāk raksturot ar citādākas tekstūras ceļa segumu, skat. 4.4. att (c), savukārt, seguma virsmas izdrupumi vairāk līdzinās bedrēm, kas ir nelielas, neregulāras un bez noteiktas formas, skatīt 4.4 att. (a, b)



(a)



(b)



(c)

#### 4.4. att. **Bedrīšu un ielāpu iezīmes** [40]

Trešā iezīmju klase ir seguma virsmas deformācijas. Rises, kas arī ietilpst pie seguma virsmas deformācijām, varētu būt viena no tām iezīmēm ar kurām neironu tīklam būtu visgrūtāk atpazīt, jo tās nereti ir grūti atšķiramas no parasta, normāla ceļa seguma, skatīt 4.4 att. (a, c). Atsevišķos leņķos rises var tikt “paslēptas” vai kā piemēram seguma virsmas viļņošanās ir grūti pamanāma ar aci, skat. 4.4. att. (b).



(a)



(b)



(c)

#### 4.3. att. **Seguma rišu un viļņošanās iezīmes** [40]

Kā redzams, tad katrai iezīmei var būt savas problemātiskās īpašības, it īpaši, ja tiek ņemti vērā visi faktori, kas ietekmē to, kā ceļš un ceļa seguma iezīmēs izskatās attēlos atkarībā no vairākiem apstākļiem:

- Video vai fotoaparātūras, t.sk., lēcas liekuma u.tml. parametriem.
- Dienas stundas jeb saules novietojuma debesīs, ēnas krituma leņķa, iezīmes dziļuma segumā.
- Laika apstākļiem, lietus laikā diezgan izteikti mainās ceļa seguma tekstūra, atsevišķas iezīmes var tikt izceltas, savukārt, citas – noslēptas, piemēram, plašu un seklu, un ar ūdeni pildītu padziļinājumu dēļ.
- Objektiem vai šķēršļiem, kas aizklāj nosakāmo iezīmi, piemēram, rudens lapas, kas nosedz seguma bojāto vietu.

Šīs iezīmes raksturo tikai melnā ceļu seguma bojājumus, kas nozīmē, ja ir nepieciešamība pēc modeļa, kas spēj noteikt bojājumus gan melnā, gan grants, gan bruģakmens ceļa seguma virsmā, tad klāt nāk daudz vairāk iezīmju un faktoru kuri ir jāņem vērā pie modeļa apmācīšanas un pielietošanas praksē.

### **4.3. Atziņas, kas iegūtas no citiem pētījumiem**

Šajā jomā ir veikti pētniecības darbi ar eksperimentālo daļu [41, 42]. Apskatot pētniecības darbu “Detecting potholes with monocular computer vision: A performance evaluation of techniques” [41] tika iegūtas dažas svarīgas atziņas:

- Bedru ir iespējams noteikt tikai kādā attāluma, jo tālāk bedres atrodas, jo grūtāk tās ir izšķiramas.
- Video vai fotoaparātūra ar lēcu, kam ir liels liekuma leņķis apgrūtina bedru noteikšanu, tāpēc ir nepieciešams veikt transformāciju, lai uzlabotu noteikšanas rezultātus.

Vēl viens svarīgs fakts, ka šī pētniecības darba [41] rezultātā tika izveidota attēlu kopa ar attēliem no braucošas mašīnas ar skatu uz ceļu. Šie attēli ir sadalīti pa grupām – tādos kuros ir bedres un tādos, kuros to nav. Šīs datu kopas izmantošana būtu labs sākuma punkts, lai apmācītu konvolūciju neironu tīklu ceļa seguma iezīmju, šajā gadījumā, bedrīšu, atpazīšanā.

Citā pētījumā [43] tiek minēts alternatīvs variants datorredzei – bedru noteikšana izmantojot telefonā iebūvēto akselerometru. Arī šādas metodes pielietošanai varētu būt potenciāls, taču arī tai ir savi trūkumi, piemēram, pie lēnas kustības akselerometra svārstības var būt nepietiekamas, lai bedri klasificētu, kā bedri, arī trajektorija pa kuru automašīna

pārvietojas uz brauktuves ir nozīme, jo šāda metodes neatradīs uz brauktuves bedres, kurām tiek pārbraukts pāri ar automašīnas korpusu nevis ar riteņiem, tādējādi neizraisot akcelerometra svārstības.

#### **4.4. Rekomendācijas tālākai izpētei un secinājumi**

Šīs tēmas padziļinātākai izpētei būtu nepieciešams veikt praktiskus eksperimentus izmantojot konvolūciju neironu tīklus, šādā veidā varētu pārliicināties par šajā darbā izvirzīto problemātisko iezīmju noteikšanas aktualitāti, tajā pat laikā būtu iespējams pārbaudīt otrajā nodaļā minēto konvolūciju neironu tīklu veidu darbību un to atšķirības.

Viens no vislielākajiem apgrūtinājumiem varētu neironu tīkla apmācīšanā izmantojot apmācības piemērus, jo attēlu kopas, kur būtu atzīmēti ceturtajā nodaļā minētie iezīmju veidi nav atrodamas, izņemot attēlu kopu, kas tika izveidota pētnieciskā darba [41] ietvaros.

Šādas problēmas risināšanai var noderēt neirona tīkla apmācīšanai tikai ar ieejas datiem, taču šis nevarētu būt pārāk labs variants, jo iezīmes uz ceļa seguma var stipri atšķirties gan pēc izmēriem, gan pēc novietojumu un citiem parametriem. Alternatīvā variantā var apsvērt mākslīgu iezīmju datu kopas ģenerēšana izmantojot ģeneratīvos neironu tīklus jeb *GAN* (angļu: *Generative Adversial Network*), šādā veidā var ģenerēt apmācības datus neironu tīklam, kas pēc tam ar tiem tiktu apmācīts.

Svarīgi arī būtu ieejas attēlu optimizācija, piemēram, platleņķa lēcas veidoti attēli var saturēt neironu tīklam lieku informāciju par apkārtni, kas nav saistīta ar ceļa segumu, tādā veidā apgrūtinot gan tīkla apmācības procesu, gan objektu noteikšanu un klasificēšanu.

Datorredzes konvolūciju neironu tīklu precizitāte objektu noteikšanā nav teicami rādītāji, kas nozīmē, ka pašbraucošo automašīnu funkcijas nodrošināšanai ir nepieciešams izmantot citas tehnoloģijas, kā piemēram – *LiDAR*, *GPS*, ultraskaņas raidītājus u.c. ierīces, kā piemēram, *Tesla* ražotajos elektrotransportlīdzekļos šīs tehnoloģija jau šodien tiek izmantotas pašbraukšanas funkcijas nodrošināšanai.

## REZULTĀTI

Darba mērķis bija apskatīt un izpētīt jaunākos datorredzes un mašīnmācīšanās risinājumus, kuriem ir izmantošanas potenciāls ceļa seguma iezīmju noteikšanā, kas ļautu atvieglot vai daļēji automatizēt ceļu uzturēšanas darbus.

Konkrētāk tika izpētīti un apskatīti mākslīgo neironu tīkli un to darbības principi, kam sekoja konvolūciju neironu tīklu uzbūves, darbības principu un to jaunāko veidu detalizētāka izpēte. Šajā darba daļā, atsaucoties uz pētniecības darbu [31] tika izdarīti secinājumi un ieteikumi konvolūciju neironu tīklu arhitektūras izvēlē atbilstoši veicamajam, ceļa seguma iezīmju noteikšanas, uzdevumam.

Neliela pētniecības darba daļa tika veltīta arī dažu populārāko mašīnmācīšanos ietvaru apskatam un nelielam salīdzinājumam un to priekšrocību un trūkumu uzskaitīšanai.

Darba noslēguma daļa tika veltīta ar ceļu seguma iezīmju noteikšanas problēmu saistītām niansēm, kuras būtu nepieciešams ņemt vērā pie konvolūciju neironu tīklu arhitektūras izveles un apmācības attēlu datu kopas izveides vai ģenerēšanas.

Vēl tika iezīmēti alternatīvie problēmas risinājumi, kas balstās uz citu datu ieguves, piemēram, no paātrinājuma sensora mobilajā telefonā, kā arī tika apskatīts un ņemtas atziņas no pētnieciskā darba, kur tika izmantota mašīnmācīšanās pieeja bedrīšu noteikšanai attēlos.

Šī pētniecības darba rezultātus var apkopot šādi:

- Reģionu bāzētais konvolūciju neironu tīkls ir piemērotāks ceļa seguma iezīmju noteikšanā, jo tas spēj atpazīt lielāku skaitu iezīmju attēlā.
- Populārākie mašīnmācīšanās ietvari balstās uz *Python* lietojumprogrammas interfeisiem (*TensorFlow*, *PyTorch*, *Caffe2*).
- Ceļa seguma iezīmes iedalās trīs superklasēs un desmit klasēs.
- Neironu tīkla apmācīšanās ļoti svarīga loma ir pareiza un korekta apmācāmās datu kopas izveidē, kas cieši saistās ar pareizu un pēc iespējas optimizētāku un objektīvāku ceļa seguma iezīmju atlasīšanu.
- Ir jāņem vērā vairāki faktori ceļa seguma iezīmju piemēru sagatavošanā un neironu tīkla darbināšanā šim uzdevumam, tie ir – apkārtējo fona tēlu minimizēšana, lēcas liekuma minimizēšana, laikapstākļu un strauji mainīgo ārējo faktoru ietekme uz neironu tīkla precizitātes apzināšana vai minimizēšana.

Darba autors uzskata, ka darba izvirzītie mērķi ir sasniegti.

## SECINĀJUMI

Pēdējo desmit gadu laikā veiktie pētījumi un atklājumi ir strauji attīstījuši datorredzes un mākslīgo neironu tīklu jomu, it īpaši sasniegumi konvolūciju neironu tīklos ir devuši vairākkārtīgus veikspējas uzlabojumus, kas ir ļāvis datorredzes problēmām rast daudz plašākus, praktiskus risinājumus.

Neironu tīklu pielietošana un izstrādāšana ir pieejama plašam interesantu lokam, jo lielākā daļa ietvaru izstrādāti pēc atvērtā koda principa. Liela daļa ietvaru izmanto *Python* programmēšanas valodu, kas pēc statistikas datiem ir viena no vispopulārākajām valodām, it īpaši eksakto zinātņu studējošo un strādājošo vidū, kas nav tiešā veidā saistīti ar datorzinātņi, piemēram, *Python* salīdzinoši vienkāršās sintakses dēļ piesaista arī matemātiķu un statistiķu mašīnmācīšanās un neironu tīklu apakšnozare.

Tika apgūts liels apjoms informācijas, kas saistīts ar mākslīgajiem neironu tīkliem, konvolūciju neironu tīkliem un to veidiem.

Neironu tīklu arhitektūru jomā ir vērojama salīdzinoši strauja izaugsme, kur pēdējo piecu gadu laikā ir bijuši izstrādāti vairāki jauni un viens par otru, atsevišķos parametros, labāki neironu tīklu modeļi.

Šo konvolūciju neironu tīklu modeļu izpēte deva priekšstatu par problēmas risinājumam nepieciešamo modeli, kā arī tīkla apmācībai un objektu noteikšanai svarīgus aspektus.

Nobeigumā var secināt, ka jaunākās, pieejamās zinātniskās literatūras pētīšana var dot labu ieskatu problēmas atrisināšanā teorētiskā līmenī, tajā pat laikā dodot norādījumus praktiskiem apsvērumiem.

Autors ir ļoti ieinteresēts pētītajā tēmā, kas pirms tam nebija tik labi pazīstama, personīgi tika uzzināts daudz jauna, kā arī veidots vispārējs priekšstats par tēmu un tās nozīmi mūsu pasaulē un nākotnes skatījumā.

## IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] “ImageNet”. [Tiešsaiste]. Pieejams internetā: <http://www.image-net.org/> [Piekļūts: 25.05.2018].
- [2] “Amazon Go: convenience and concern at new checkout-free corner shop”, The Guardian. [Tiešsaiste]. Pieejams internetā: <https://www.theguardian.com/us-news/2018/jan/22/amazon-go-convenience-store-corner-shop> [Piekļūts: 25.05.2018].
- [3] “Tesla Autopilot” [Tiešsaiste]. Pieejams internetā: <https://www.tesla.com/autopilot> [Piekļūts: 25.05.2018].
- [4] “Tesla driver dies in first fatal crash while using autopilot mode”, The Guardian. [Tiešsaiste]. Pieejams internetā: <https://www.theguardian.com/technology/2016/jun/30/tesla-autopilot-death-self-driving-car-elon-musk> [Piekļūts: 25.05.2018].
- [5] “Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian”, The Guardian. [Tiešsaiste]. Pieejams internetā: <https://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe> [Piekļūts: 25.05.2018].
- [6] “Video shows Tesla autopilot failing at the fatal crash site”, CBS News. [Tiešsaiste]. Pieejams internetā: <https://www.cbsnews.com/video/video-shows-tesla-autopilot-failing-at-site-of-fatal-march-crash/> [Piekļūts: 25.05.2018].
- [7] “Valsts autoceļu tīkla statistika, 2016”, VAS Latvijas Valsts ceļi. 2016. [Tiešsaiste]. [https://lvceli.lv/wp-content/uploads/2015/08/Latvijas\\_Valsts\\_Celi\\_2016\\_115x118\\_20170719\\_oranzs.pdf](https://lvceli.lv/wp-content/uploads/2015/08/Latvijas_Valsts_Celi_2016_115x118_20170719_oranzs.pdf) [Piekļūts: 25.05.2018].
- [8] “CS231n Convolutional Neural Networks for Visual Recognition”, Stanford University., 2018. [Tiešsaiste]. Pieejams internetā: <http://cs231n.github.io/neural-networks-1/> [Piekļūts: 25.05.2018].
- [9] Michael London and Michael Häusser, “Dendritic Computation”, *Annual Review of Neuroscience*, vol. 28, 503.-533. lpp., 2005. [Tiešsaiste]. Pieejams internetā: [https://neurophysics.ucsd.edu/courses/physics\\_171/annurev.neuro.28.061604.135703.pdf](https://neurophysics.ucsd.edu/courses/physics_171/annurev.neuro.28.061604.135703.pdf) [Piekļūts: 25.05.2018].

- [10] Jānis Zuters, “Neironu Tīkli”. [Tiešsaiste]. Pieejams internetā:  
<http://home.lu.lv/~janiszu/courses/eanns/eanns.pdf> [Piekļūts: 25.05.2018].
- [11] Eric Roberts, “History: The 1940's to the 1970's”, Stanford University. [Tiešsaiste].  
Pieejams internetā:  
<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html> [Piekļūts: 25.05.2018].
- [12] Eric Roberts, “History: The 1980's to the present”, Stanford University. [Tiešsaiste].  
Pieejams internetā:  
<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html> [Piekļūts: 25.05.2018].
- [13] Simon Haykin, “Neural Networks and Learning Machines”, Pearson Education, Upper Saddle River, NJ, 3rd edition, 906 lpp., 2009. [Tiešsaiste]. Pieejams internetā:  
<http://dai.fmph.uniba.sk/courses/NN/haykin.neural-networks.3ed.2009.pdf> [Piekļūts: 25.05.2018].
- [14] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ”ImageNet Classification with Deep Convolutional Neural Networks”, *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, vol 1, 1097-1105 lpp., 2012. [Tiešsaiste]. Pieejams internetā:  
<http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf> [Piekļūts: 25.05.2018]
- [15] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, Yoshua Bengio, “Maxout Networks”, *ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning*, vol 28, section 3, 1319. – 1327. lpp., 2013. [Tiešsaiste]. Pieejams internetā:  
<http://proceedings.mlr.press/v28/goodfellow13.pdf> [Piekļūts: 25.05.2018]
- [16] Ian J. Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 800 lpp., 2016. [Tiešsaiste]. Pieejams internetā: <http://www.deeplearningbook.org/>  
[Piekļūts: 25.05.2018]
- [17] “Image filtering, Convolution”, Machine Learning Guru. [Tiešsaiste]. Pieejams internetā:  
[http://machinelearningguru.com/computer\\_vision/basics/convolution/image\\_convolution\\_1.html](http://machinelearningguru.com/computer_vision/basics/convolution/image_convolution_1.html) [Piekļūts: 25.05.2018]

- [18] Dhruv Parhasarathy, “A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN”, 2017. [Tiešsaiste]. Pieejams internetā: <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4> [Piekļūts: 25.05.2018]
- [19] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, 21 lpp., 2013. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1311.2524> [Piekļūts: 25.05.2018]
- [20] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, “Selective Search for Object Recognition”, 14 lpp., 2012. [Tiešsaiste]. Pieejams internetā: <https://www.koen.me/research/pub/uijlings-ijcv2013-draft.pdf> [Piekļūts: 25.05.2018]
- [21] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, 16 lpp., 2013. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1312.6229> [Piekļūts: 25.05.2018]
- [22] Ross Girshick, “Fast R-CNN”, 9 lpp., 2015. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1504.08083> [Piekļūts: 25.05.2018]
- [23] Karen Simonyan, Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, 2014. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1409.1556> [Piekļūts: 25.05.2018]
- [24] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 14 lpp., 2016. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1506.01497> [Piekļūts: 25.05.2018]
- [25] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, “Mask R-CNN”, 12 lpp., 2017. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1703.06870> [Piekļūts: 25.05.2018]
- [26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector”, 17 lpp., 2015. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1512.02325>

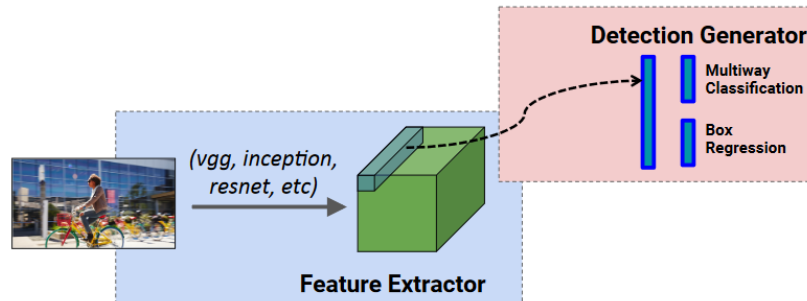
- [Piekļūts: 25.05.2018]
- [27] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, 10 lpp., 2015. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1506.02640> [Piekļūts: 25.05.2018]
- [28] Jonathan Hui, “SSD object detection: Single Shot MultiBox Detector for real-time processing”, 2018. [Tiešsaiste]. Pieejams internetā: [https://medium.com/@jonathan\\_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06](https://medium.com/@jonathan_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06) [Piekļūts: 25.05.2018]
- [29] Jonatahn Hui, “What do we learn from region based object detectors (Faster R-CNN, R-FCN, FPN)?”, 2018. [Tiešsaiste]. Pieejams internetā: [https://medium.com/@jonathan\\_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9](https://medium.com/@jonathan_hui/what-do-we-learn-from-region-based-object-detectors-faster-r-cnn-r-fcn-fpn-7e354377a7c9) [Piekļūts: 25.05.2018]
- [30] Jonatahn Hui, “What do we learn from single shot object detectors (SSD, YOLOv3), FPN & Focal loss (RetinaNet)?”, 2018. [Tiešsaiste]. Pieejams internetā: [https://medium.com/@jonathan\\_hui/what-do-we-learn-from-single-shot-object-detectors-ssd-yolo-fpn-focal-loss-3888677c5f4d](https://medium.com/@jonathan_hui/what-do-we-learn-from-single-shot-object-detectors-ssd-yolo-fpn-focal-loss-3888677c5f4d) [Piekļūts: 25.05.2018]
- [31] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, Kevin Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors”, 21 lpp., 2016. [Tiešsaiste]. Pieejams internetā: <https://arxiv.org/abs/1611.10012> [Piekļūts: 25.05.2018]
- [32] “Comparing Top Deep Learning Frameworks: Deeplearning4j, PyTorch, TensorFlow, Caffe, Keras, MxNet, Gluon & CNTK” [Tiešsaiste]. Pieejams internetā: <https://deeplearning4j.org/compare-dl4j-tensorflow-pytorch> [Piekļūts: 25.05.2018]
- [33] “TensorFlow” [Tiešsaiste]. Pieejams internetā: <https://www.tensorflow.org/> [Piekļūts: 25.05.2018]

- [34] Indra den Bakker, “Battle of the Deep Learning frameworks—Part I: 2017, even more frameworks and interfaces”, 2017. [Tiešsaiste]. Pieejams internetā: <https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750> [Piekļūts: 25.05.2018]
- [35] “Caffe” [Tiešsaiste]. Pieejams internetā: <http://caffe.berkeleyvision.org/> [Piekļūts: 25.05.2018]
- [36] “Caffe 2” [Tiešsaiste]. Pieejams internetā: <https://caffe2.ai/> [Piekļūts: 25.05.2018]
- [37] “Torch” [Tiešsaiste]. Pieejams internetā: <http://torch.ch/> [Piekļūts: 25.05.2018]
- [38] “PyTorch” [Tiešsaiste]. Pieejams internetā: <https://pytorch.org/> [Piekļūts: 25.05.2018]
- [39] Kenneth Tran, “Evaluation of Deep Learning Toolkits”, 2016. [Tiešsaiste]. Pieejams internetā: <https://github.com/zer0n/deepframeworks/blob/master/README.md> [Piekļūts: 25.05.2018]
- [40] Viktors Haritonovs, Jānis Tihonovs, Artūrs Riekstiņš, Rolands Īzaks, Edgars Eglītis, “Asfalta maisījuma noturība pret plastiskā deformācijām. Plānkārtas ceļa segas dilumkārtas slāņu (BBTM) un citu bituminēto segumu atjaunošanas un pārbūves tehnoloģiju izpēte”, VAS Latvijas Valsts ceļi, 88 lpp., 2016. [Tiešsaiste]. Pieejams internetā: [https://lvceļi.lv/wp-content/uploads/2015/02/Petijums\\_Plankartas\\_cela\\_segas\\_dilumkartas\\_slanu\\_BBTM\\_un\\_citu\\_bitumin.pdf](https://lvceļi.lv/wp-content/uploads/2015/02/Petijums_Plankartas_cela_segas_dilumkartas_slanu_BBTM_un_citu_bitumin.pdf) [Piekļūts: 25.05.2018]
- [41] S. Nienaber, M.J. Booysen, R.S. Kroon, “Detecting potholes using simple image processing techniques and real-world footage”, SATC, Pretoria, South Africa, 2015. [Tiešsaiste]. Pieejams internetā: <http://scholar.sun.ac.za/handle/10019.1/97191> [Piekļūts: 25.05.2018]
- [42] S. Nienaber, R.S. Kroon, M.J. Booysen, “A Comparison of Low-Cost Monocular Vision Techniques for Pothole Distance Estimation”, IEEE CIVTS, Cape Town, South Africa, 2015. [Tiešsaiste]. Pieejams internetā: <https://researchspace.csir.co.za/dspace/handle/10204/8801> [Piekļūts: 25.05.2018]

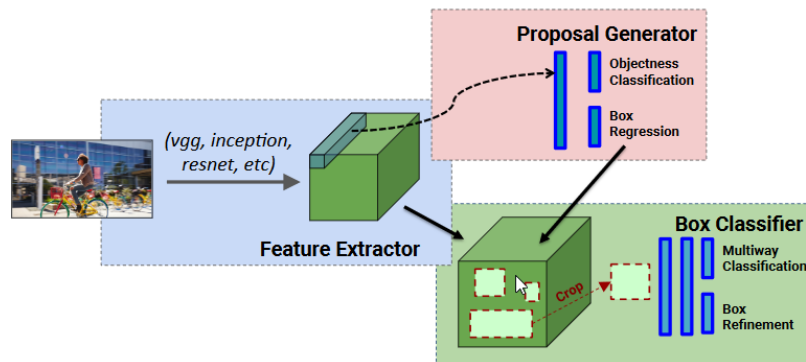
- [43] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, Leo Selavo, “Real Time Pothole Detection using Android Smartphones with Accelerometers”, 2011. [Tiešsaiste]. Pieejams internetā:  
<http://strazdins.lv/papers/mednis2011androidpotholes.pdf> [Pieklūts: 25.05.2018]

# PIELIKUMI

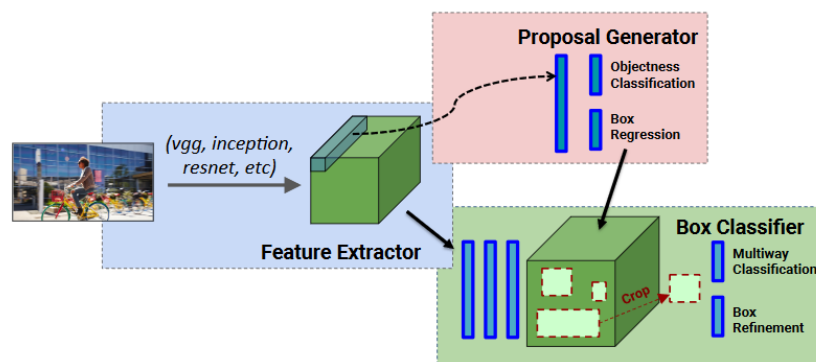
## 1. Pielikums. CNN arhitektūru vispārējās uzbūves kopsavilkums



(a) SSD.



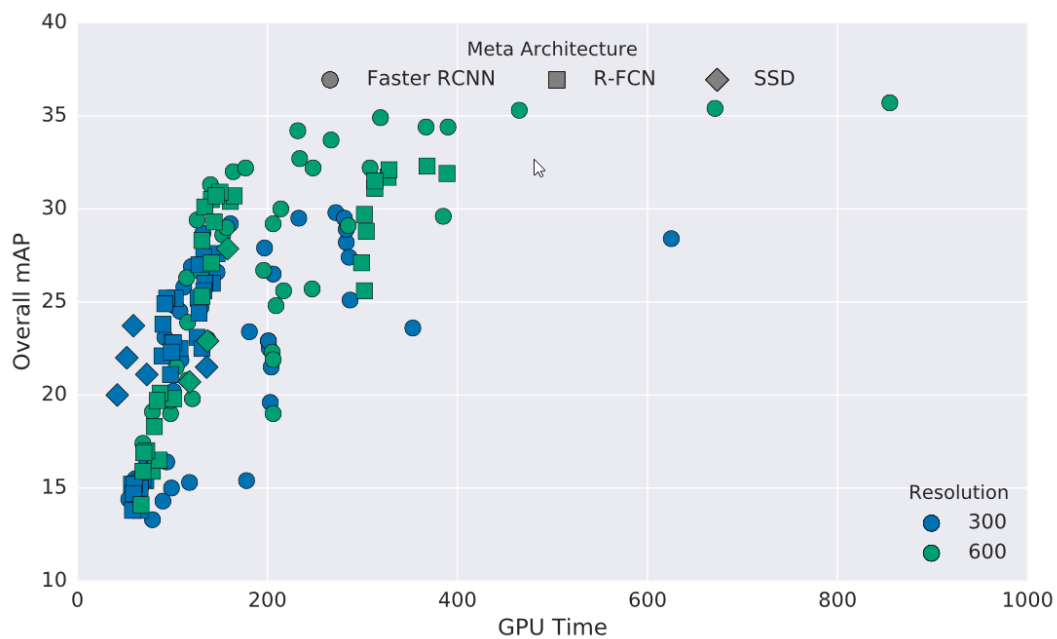
(b) Faster RCNN.



(c) R-FCN.

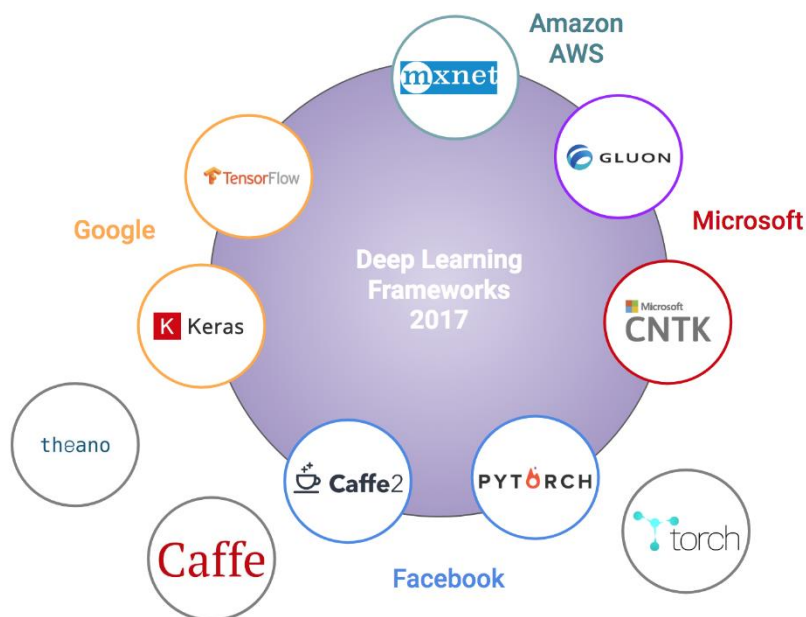
Avots: [31]

## 2. Pielikums. Konvolūciju neironu tīklu veidu ātrdarbības salīdzinājums atkarībā no attēla izšķirtspējas



Avots: [31]

## 3. Pielikums. Mašīnmācīšanās ietvaru kopskats, 2017. gads



Avots: [34]

#### 4. Pielikums. Seguma virsmas novērtēšanas tabula Ontario, Kanāda

		Svars seguma atjaunošanās stratēģijā	Bojājumu pakāpe (Si)					Bojāto vietu blīvums (Di)					
			Ļoti zema	Zema	Vidēja	Liela	Ļoti liela	Zems	Vidējs	Biežs	Plašs	Viscaur	
								< 10%	10 - 20	20 - 40	40 - 80	> 80%	
Seguma bojājumi		<b>W<sub>i</sub></b>	1	2	3	4	5	1	2	3	4	5	
Virsmas defekti	Izdrupšana ( <i>Ravelling</i> )	3.0		x					x				
	Svišana ( <i>Bleeding</i> )	1.5		x					x				
Virsmas deformācijas	Stumšanās ( <i>Shoving</i> )	1.0	x					x					
	Rises	3.0	x					x					
	Sagrozīšanās ( <i>distortion</i> )	3.0		x					x				
Plaisas	Garenplaisa riteņa sliežu vietā	Viena vai vairākas	1.5	x					x				
		Tīklveida ( <i>Alligator</i> )	3.0	x					x				
	Garenassplaisa	Viena vai vairākas	0.5			x					x		
		Tīklveida	2.0	x					x				
	Apmalles plaisa	Viena vai vairākas	0.5			x					x		
		Tīklveida	1.5			x					x		
	Šķersplaisa	Puse garuma, pilna garuma, vairākas	1.0	x					x				
		Tīklveida	3.0	x					x				
	Likumainā garenplaisa		1.0		x					x			
	Haotiska		0,5										

2.3. att. Seguma virsmas novērtēšanas tabula Ontario, Kanādā [7]

Avots: [40]

Bakalaura darbs “Ceļa seguma iezīmju noteikšana, izmantojot mašīnmācīšanos” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: \_\_\_\_\_ / Krišjānis Šulcs /

Rekomendēju/nerekomendēju darbu aizstāvēšanai (nevajadzīgo izsvītrot)

Vadītājs: vecākais programmētājs, M. Dat. Muntis Rudzītis: \_\_\_\_\_

Recenzents: zinātniskais asistents, M. Dat. Ilvars Mizniks

Darbs iesniegts Datorikas fakultātē 28.05.2018.

Dekāna pilnvarotā persona

Vecākā metodiķe Ārija Sproģe: \_\_\_\_\_

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

Komisija: