

## DEEP LEARNING FOR APPLE AND PEAR RECOGNITION DZIĻĀ APMĀCĪBA ĀBOLU UN BUMBIERU ATPAZĪŠANAI

Authors: **Vitālijs ŽUKOVŠ**, vz16020@edu.rta.lv; **Ilmārs APEINĀNS**, ia16024@edu.rta.lv  
Scientific supervisor: **Sergejs KODORS**, Dr.sc.ing. sergejs.kodors@rta.lv  
Rezekne Academy of Technologies, Atbrīvošanas aleja 115, Rēzekne, Latvia

---

**Abstract.** The aim of this work is to develop a neural network, which can recognize apples and pears. To achieve the goal, the authors applied AlexNet architecture and the open dataset "Fruits360". The trained model showed a good result testing it on validation images - total accuracy 0.97 and latency 35ms/step. In the future research, authors consider training the neural network model using the MobileNet architecture and verify it using the Cohen's Kappa coefficient.

**Keywords.** AlexNet, apple, CNN, Fruits360, Food2030, neural network, pear

---

### Ievads

*Transfer learning* ir mašīnmācīšanas metode, kad modelis, kurš ir izstrādāts priekš viena uzdevuma, tiek atkārtoti izmantots citā uzdevumā [2]. Šī ir populāra pieeja dziļajās apmācībā, kur iepriekš sagatavoti modeļi tiek izmantoti kā sagatave datora redzes un dabiskās valodas apstrādes uzdevumiem, ņemot vērā milzīgos aprēķina un laika resursus, kas nepieciešami neironu tīkla modeļu izstrādāšanai.

Neironu tīkla apmācīšana ietver sevi arhitektūru, datu kopu, apmācības parametrus un datu kopas pārveidojumus. Izmantojot AlexNet arhitektūru, CIFAR-10, CIFAR-100 [6] un Fruits 360 [3] datu kopas, tika apmācīts neironu tīkls, pielietojot *transfer learning* metodi (tālāk TL).

Šī pētījuma mērķis ir apmācīt neironu tīklu, pielietojot TL metodi un salīdzināt rezultātus ar neironu tīklu bez TL metodes pielietošanas.

### 1. Materiāli un metodes

Eksperimenta laikā tika pielietotas trīs datu kopas: CIFAR-10, CIFAR-100 un Fruits360. Datu kopu īpašības ir attēlotas 1. tabulā.

Lai izpildītu mašīnāpmācības uzdevumus, tika pielietoti TensorFlow un Keras izstrādes rīki. TensorFlow [4] korektam darbam ar GPU, bija nepieciešams pieinstalēts klāt:

- Python 3.7.6 – programmēšanas valoda;
- CUDA Toolkit – paralēlas skaitļošanas platforma neiron tīklā paātrināšanai izmantojot GPU jaudu;
- cuDNN – dziļās apmācības bibliotēka, kura nodrošina optimizēšanu standarta operācijām: apvienošana, normalizācija un slāņu aktivācija;
- Anaconda – Python datu zinātnes standarta platforma, kura ir paredzēta mašīnmācīšanas uzdevumiem [5];
- TensorFlow 2.1.0 – atvērta bibliotēka mašīnāpmācības uzdevumiem, kuru izstrādāja Google, lai ātri realizētu neiron tīklus un apmācītu tos.
- TensorFlow GPU – tiek izmantota veikt aprēķinus pielietojot GPU.

Datu kopas

Nosaukums	Klases	Attēli	Trenēšanai	Pārbaudei
<i>Fruits 360</i>	120	81120	60498	20622
<i>CIFAR-10</i>	10	60000	50000	10000
<i>CIFAR-100</i>	100	60000	50000	10000

**Eksperimenta** laikā tika sagatavots *Python* skripts, kuru var sadalīt uz 6. posmiem:

1. Datu kopu sagatavošana un datu augmetācija [7, 8];
2. Modeļa apmācība ar *CIFAR-10* un *CIFAR-100*;
3. Modeļa slāņu iesaldēšana un izejas mainīšana;
4. Modeļa apmācība ar *Fruits360-120*;
5. Modeļa slāņu iesaldēšana un izejas mainīšana;
6. Modeļa apmācība ar *Fruits360-3*.

Datu sagatavošanā ietilpst datu kopas atrašana, pieslēgšana un attēlu sakārtošana. Tālāk sakārtoti dati tiek apstrādāti (pagriezti, palielināti, utt.). Datu augmentācija ir metode, kā mākslīgi izveidot jaunus apmācības datus no esošajiem apmācības datiem. Tas tiek darīts, piemērojot specifiskās metodes piemēriem no apmācības datiem, kas rada jaunus un atšķirīgus apmācības piemērus.

Otrajā punkta modelis tiek apmācīts ar *CIFAR* datu kopas palīdzību. Pirmais slānis tiek iesaldēts, tāpēc ka tas atbild par līniju, krāsu, leņķu atpazīšanu (skat. *1.1. att.*) un izmainīta izeja, no 10 un 100 klasēm uz 120.



*1.1. attēls. Modeļa iesaldēto slāņu vizualizācijas piemērs [1]*

Tālāk modelis tiek apmācīts atpazīt augļus ar *Fruits360* datu kopas palīdzību un beigās tiek nomainīts klašu daudzums no *Fruits360-120* uz *Fruits360-3*, kas ir nepieciešams tieši priekš ābolu, bumbieru attēlu atpazīšanas. Modeļa apmācība ar *Fruits360-3* tika atkārtota 10 reizes.

Sākumā, modelis tiek apmācīts atpazīt līnijas un krāsas. Ar *Fruits360-120* datu kopu, modelis tiek apmācīts atpazīt augļus un pēc tam to pārāpmāca atpazīt tieši tos augļus, kuri ir nepieciešami (ābolus un bumbierus).

Eksperimentam par pamatu tika paņemta *AlexNet* arhitektūra. Eksperimenta *AlexNet* arhitektūras modeli, kas tika izmēģināti eksperimentā ir attēloti *2. tabulā*. Izmantotajiem arhitektūras modeļiem ir dažādi filtru izmēri. Filtra izmērs iestata konvolūcijas darbībā izmantoto filtru skaitu.

Kā var pamanīt, agrākie 2D konvolūcijas slāņi, kas ir tuvāk ievadei, mācās mazāk filtru, bet vēlākie - vairāk.

## Eksperimenta AlexNet modeli

<i>Layer (type)</i>	<i>AlexNet16</i>	<i>AlexNet32</i>	<i>AlexNet64</i>
<i>conv2d-1</i>	(48, 48, 16)	(48, 48, 32)	(48, 48, 64)
<i>batch_normalization</i>	(48, 48, 16)	(48, 48, 32)	(48, 48, 64)
<i>max_pooling2d</i>	(24, 24, 16)	(24, 24, 32)	(24, 24, 64)
<i>conv2d-2</i>	(20, 20, 16)	(20, 20, 32)	(20, 20, 64)
<i>batch_normalization-1</i>	(20, 20, 16)	(20, 20, 32)	(20, 20, 64)
<i>average_pooling2d</i>	(10, 10, 16)	(10, 10, 32)	(10, 10, 64)
<i>dropout-1</i>	(10, 10, 16)	(10, 10, 32)	(10, 10, 64)
<i>conv2d-3</i>	(7, 7, 32)	(7, 7, 64)	(7, 7, 128)
<i>batch_normalization-2</i>	(7, 7, 32)	(7, 7, 64)	(7, 7, 128)
<i>dropout</i>	(7, 7, 32)	(7, 7, 64)	(7, 7, 128)
<i>dense</i>	(512)	(512)	(512)
<i>dropout-2</i>	(512)	(512)	(512)
<i>out_layer</i>	(10)	(10)	(10)

## 2. Rezultāti

Neironīklu apmācības rezultāti ir apkopoti 3.-8. tabulās. 3. un 4. tabulā ir attēloti AlexNet16 apmācības un modeļa parametri, kā arī eksperimenta rezultāti, izmantojot CIFAR datu kopas priekšapmācības laikā. Atbilstošu informāciju var atrast par AlexNet32 (skat. 5.-6. tabulas) un AlexNet64 (skat. 7.-8. tabulas).

## AlexNet16

<i>CIFAR-10</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
524	200	605	1056	956 426	96	0.82686	0.7872
<i>FRUITS360-120</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
100	20	795	320	988 504	24 448	0.95552	0.97963
<i>FRUITS360-3 (average of 10 repeat)</i>							
<i>Time (min)</i>			<i>Accuracy</i>			<i>Validation accuracy</i>	
3.28			0.96269			0.97239	

## AlexNet16

<i>CIFAR-100</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
763	200	571	1449	1 002 596	96	0.47080	0.48100
<i>FRUITS360-120</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
100	20	800	310	988 504	24, 448	0.98681	0.96856
<i>FRUITS360-3 (average of 10 repeat)</i>							
<i>Time (min)</i>			<i>Accuracy</i>			<i>Validation accuracy</i>	
8.53			0.96474			0.97514	

5. tabula

*AlexNet32*

<b>CIFAR-10</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
573	200	623	1212	1 672 426	256	0.82126	0.7879
<b>FRUITS360-120</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
74	20	806	240	1 726 424	2 688	0.99126	0.99695
<b>FRUITS360-3 (average of 10 repeat)</b>							
<i>Time (min)</i>			<i>Accuracy</i>			<i>Validation accuracy</i>	
8.91			0.96643			0.95765	

6. tabula

*AlexNet32*

<b>CIFAR-100</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
366	200	560	683	1 718 596	256	0.56978	0.49260
<b>FRUITS360-120</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
73	20	780	227	1 726 424	2 688	0.99246	0.99622
<b>FRUITS360-3 (average of 10 repeat)</b>							
<i>Time (min)</i>			<i>Accuracy</i>			<i>Validation accuracy</i>	
5.25			0.97208			0.95673	

7. tabula

*AlexNet64*

<b>CIFAR-10</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
390	200	645	845	4 012 618	256	0.88224	0.7793
<b>FRUITS360-120</b>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
63	20	847	211	4 064 184	5 120	0.99567	0.99598
<b>FRUITS360-3 (average of 10 repeat)</b>							
<i>Time (min)</i>			<i>Accuracy</i>			<i>Validation accuracy</i>	
10.27			0.97319			0.94981	

*AlexNet64*

<i>CIFAR-100</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
306	200	566	576	4 058 788	256	0.67250	0.48950
<i>FRUITS360-120</i>							
<i>Epoch</i>	<i>Sleep</i>	<i>Step time (ms/step)</i>	<i>Time (min)</i>	<i>Trainable params</i>	<i>Non-trainable params</i>	<i>Accuracy</i>	<i>Validation accuracy</i>
50	20	761	153	4 027 256	42 048	0.99076	0.99525
<i>FRUITS360-3 (average of 10 repeat)</i>							
<i>Time (min)</i>			<i>Accuracy</i>		<i>Validation accuracy</i>		
5.46			0.94406		0.93027		



1.2. attēls. Modeļa trenēšanas un validācijas diagrammas piemērs

**Secinājumi**

Neironu tīklu apmācībā ir diezgan liels un apjomīgs laika ziņā process. Priekš apmācības ir nepieciešams sagatavot datu kopu, ka arī izvēlēties arhitektūru. Pielietojot *transfer learning* metodi, ir iespējams atkārtoti izmantot iepriekš apmācītu modeli citām vajadzībām. Izmantojot *AlexNet* arhitektūru un *CIFAR* un *FRUITS360* datu kopas tika veikts *transfer learning* eksperiments, kurā, ar *CIFAR* datu kopas apmācītais modelis, tika pārmācīts atpazīt ābolus un bumbierus ar *FRUITS360* datu kopas palīdzību.

Eksperiments tika sadalīts uz trim etapiem:

1. Modeļa apmācība ar *CIFAR10* aizņēma ~17,5 stundas un rezultātā tika iegūta apmācības precizitāte ~0.78% (priekš *AlexNet16* arhitektūras);
2. Modeļa apmācība ar *FRUITS360-120* aizņēma ~5.3 stundas un rezultātā tika iegūta precizitāte ~0.98% (priekš *AlexNet16* arhitektūras);
3. Modeļa apmācība ar *FRUITS360-3* aizņēma ~3.5 minūtes un tika iegūta precizitāte ~0.96% (priekš *AlexNet16* arhitektūras).

Līdz ar to var secināt, ka apmācības laiks ir samazinājies 3 reizēs.

## **Acknowledgement**

*Funding institution: Latvian Council of Science*

*Funding number: lzp-2019/1-0094*

*Acronym: FLPP-2019-1*

*Funding text: This research is funded by the Latvian Council of Science, project "Application of deep learning and datamining for the study of plant-pathogen interaction: the case of apple and pear scab", project No. lzp-2019/1-0094*

## **Summary**

*Neural network training is a rather large and time-consuming process. For training it is necessary to prepare a data set, as well as to choose the architecture. Using the transfer learning method, it is possible to reuse a previously trained model for other purposes. Using the AlexNet architecture and the CIFAR-FRUIT360 data set, a transfer learning experiment was performed in which, with the CIFAR data set, the model was trained to recognize apples and pears using the FRUIT360 data set.*

*The experiment was divided into three stages:*

- 1. Model training with CIFAR10 took ~17.5 hours and as a result training accuracy ~0.78% was obtained (for AlexNet16 architecture);*
- 2. Model training with FRUIT360-120 took ~5.3 hours and resulted in accuracy of ~0.98% (for AlexNet16 architecture);*
- 3. Model training with FRUIT360-3 took ~3.5 minutes and the accuracy was ~0.96% (for AlexNet16 architecture).*

*Thus, it can be concluded that the training time has decreased 3 times.*

## **Literatūra**

[1] Konvolūcijas neironu tīkli attēlu atpazīšanai [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://habr.com/ru/post/456186/>

[2] *Transfer learning* [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://machinelearningmastery.com/transfer-learning-for-deep-learning/>

[3] *Fruits360* datu kopa [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://www.kaggle.com/moltean/fruits>

[4] *TensorFlow* [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://www.tensorflow.org/about>

[5] *Anaconda* rīks [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://www.anaconda.com/why-anaconda/>

[6] CIFAR datu kopa [tiešsaiste], [atsauce uz 01.03.2021]. Pieejams:

<https://www.cs.toronto.edu/~kriz/cifar.html>

[7] Datu augmentācija [tiešsaiste], [atsauce uz 03.03.2021]. Pieejams:

<https://algorithmia.com/blog/introduction-to-dataset-augmentation-and-expansion>

[8] Datu augmentācija [tiešsaiste], [atsauce uz 07.03.2021]. Pieejams:

<https://medium.com/analytics-vidhya/data-augmentation-is-it-really-necessary-b3cb12ab3c3f>