

Seminár Robotika.SK

Rozpoznávanie tváre pomocou hlbokého učenia

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

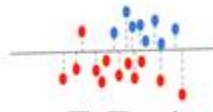
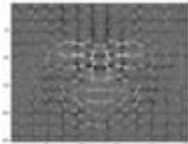
http://dai.fmph.uniba.sk/w/Andrej_Lucny

www.robotika.sk/seminar/2019/cviko9-win.zip

Face detector



HOG

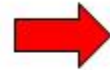
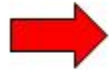


LDA

Facial landmark detector



Cascade regressor



Face image



The face image is divided into blocks

LBP

1	2	2
9	5	6
5	3	1

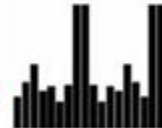
Threshold

0	0	0
1		1
1	0	0

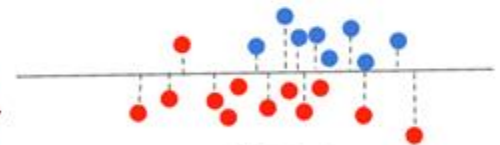
Binary: 00010011
Decimal: 19



LBP histogram from each block



Feature histogram

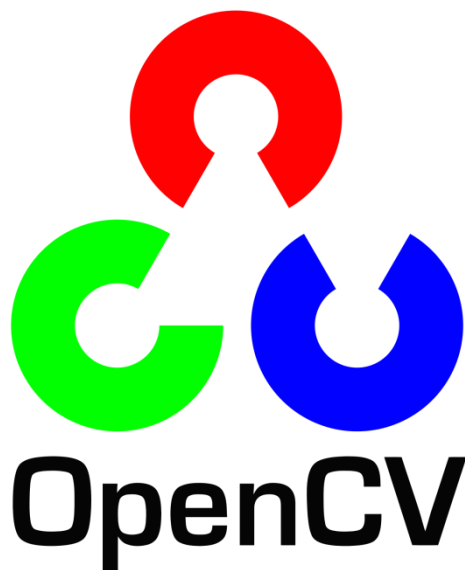


LDA

“Andy”

LBPH – Local binary pattern histogram

Klasické riešenie

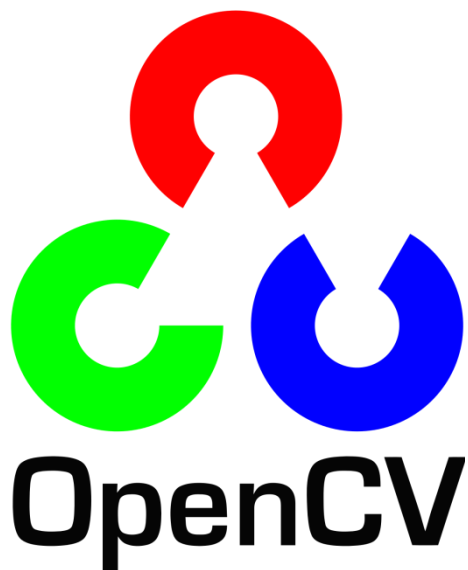


4.0.1

- Open source knižnica na počítačové videnie
- Intel, Itseez (Gary Bradski)
- BSD Licencia
- 3000 rôznych algoritmov
- C++, Java, Python
- Windows, Linux, Android
- Je to fakt rýchle spoľahlivé
- Podpora CUDA a OpenCL
- **Podpora ML a DNN**

od 11/2018

Veľa pekných zdrojákov sa dá získať za registráciu zdarma z článkov na www.learnopencv.com a ich projekte na GitHub-e



4.0.1

- <https://github.com/opencv/opencv>
- https://github.com/opencv/opencv_contrib

od 11/2018 • https://github.com/opencv/open_model_zoo

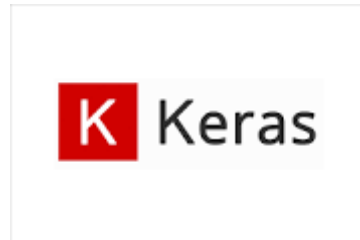


19.16

- Open source knižnica na strojové učenie (vrátane **deep learning**) a počítačové videnie
- Autor: Davis King
- Boost Software License
- Linux, Windows, Mac
- CUDA based
- Od roku 2002

Deep Learning na GPU

Caffe



PyTorch



GLUON



mxnet

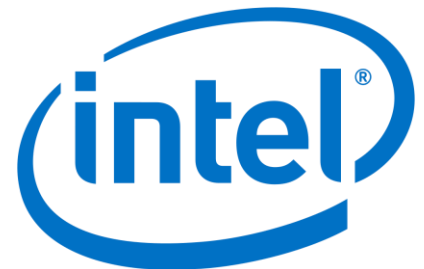


Deep Learning na CPU

OpenVINO™

C++

 python™



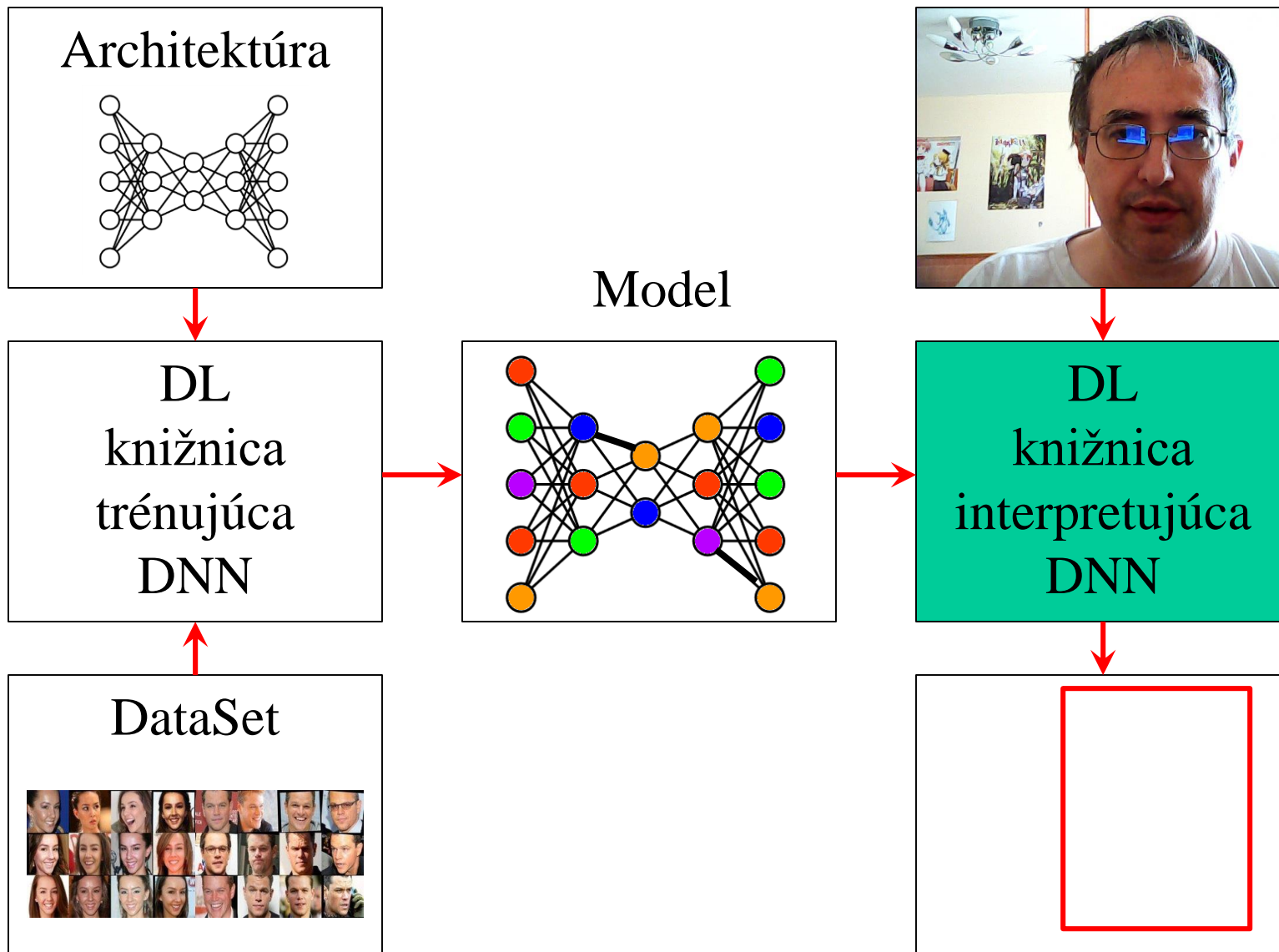
Model ZOO

- VGG
- ResNet
- AlexNet
- DarkNet
- GoogleNet
- ...
- *Gluon*

DataSets

- Mnist
- CIFAR
- ImageNet
- Kaggle
- UC Berkeley
- ...

Použite Deep Learning-u



Face detector



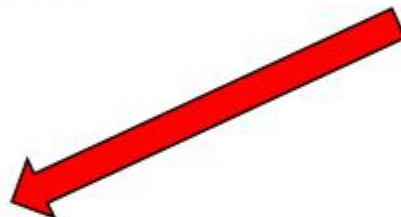
ResNet



Facial landmark detector



Cascade
regressor



ResNet



Descriptor

$(0.453, 0.122, 0.998, \dots)$

Enrollment

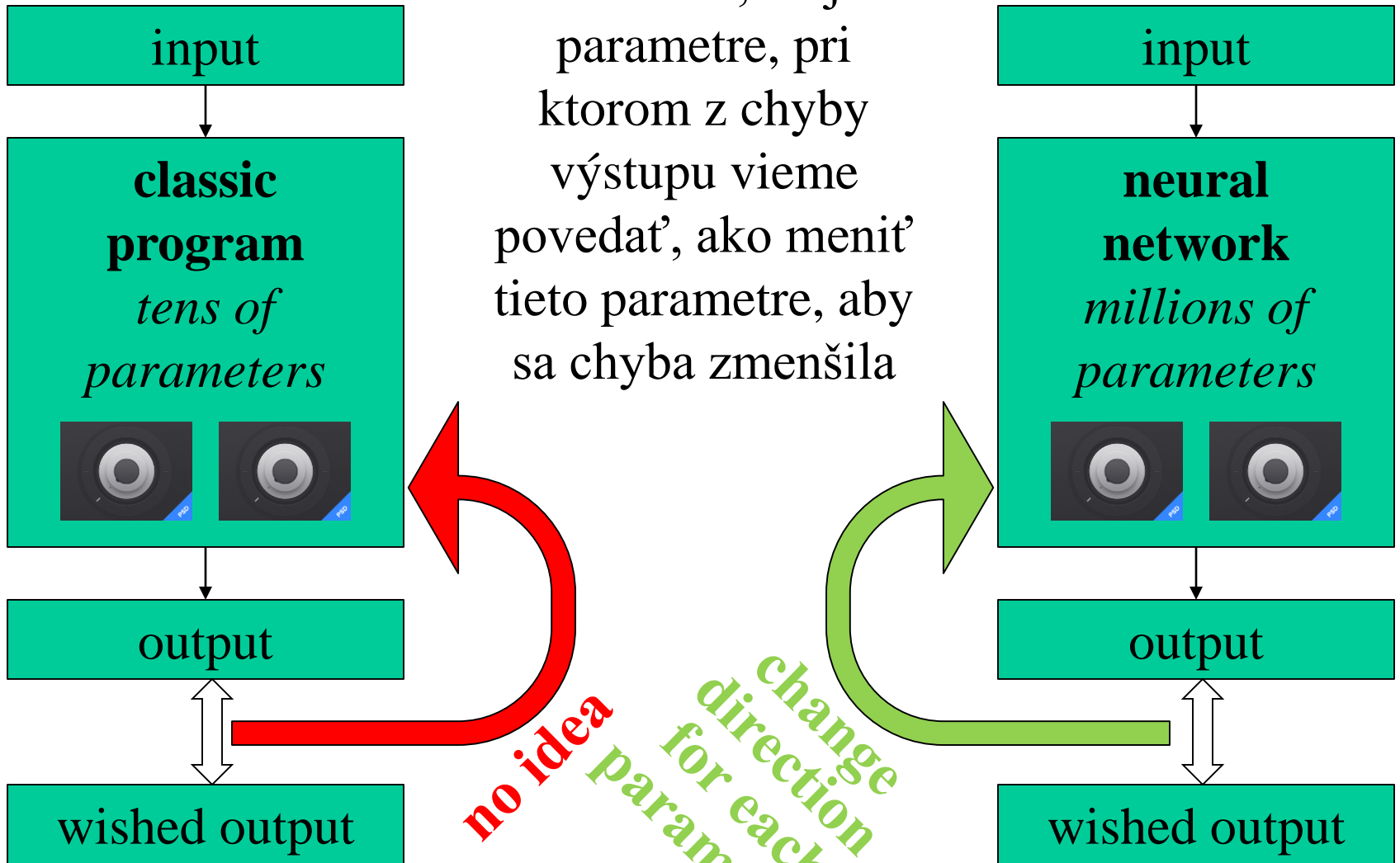


“Andy”

Riešenie na báze DL

Neurónová sieť

= čokoľvek, majúce
parametre, pri
ktorom z chyby
výstupu vieme
povedať, ako meniť
tieto parametre, aby
sa chyba zmenšila



DNN – hlboká neurónová sieť

- Má veľa vrstiev (desiatky desiatok až stoviek)
- Obsahuje konvolučné vrstvy
- Odoláva problému miznúceho gradientu (vanishing gradient) a ďalších príčin preučenia (overfitting)

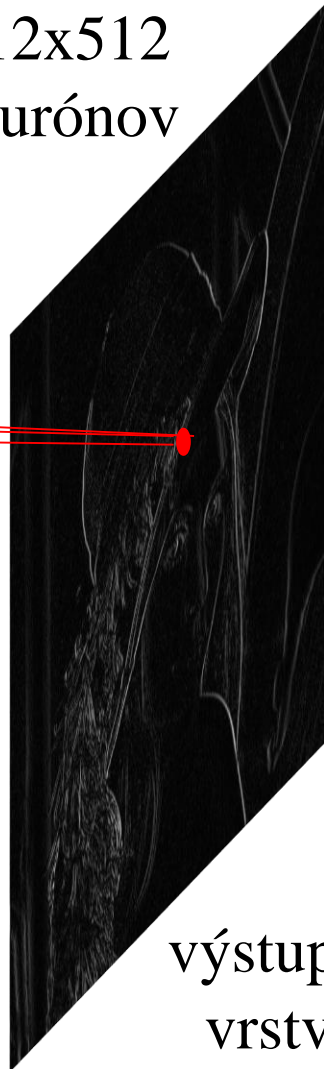
Konvolučná neurónová sieť (príklad): hranový operátor

512x512
neurónov



vstupná
vrstva

512x512
neurónov



výstupná
vrstva

kernel 3x3

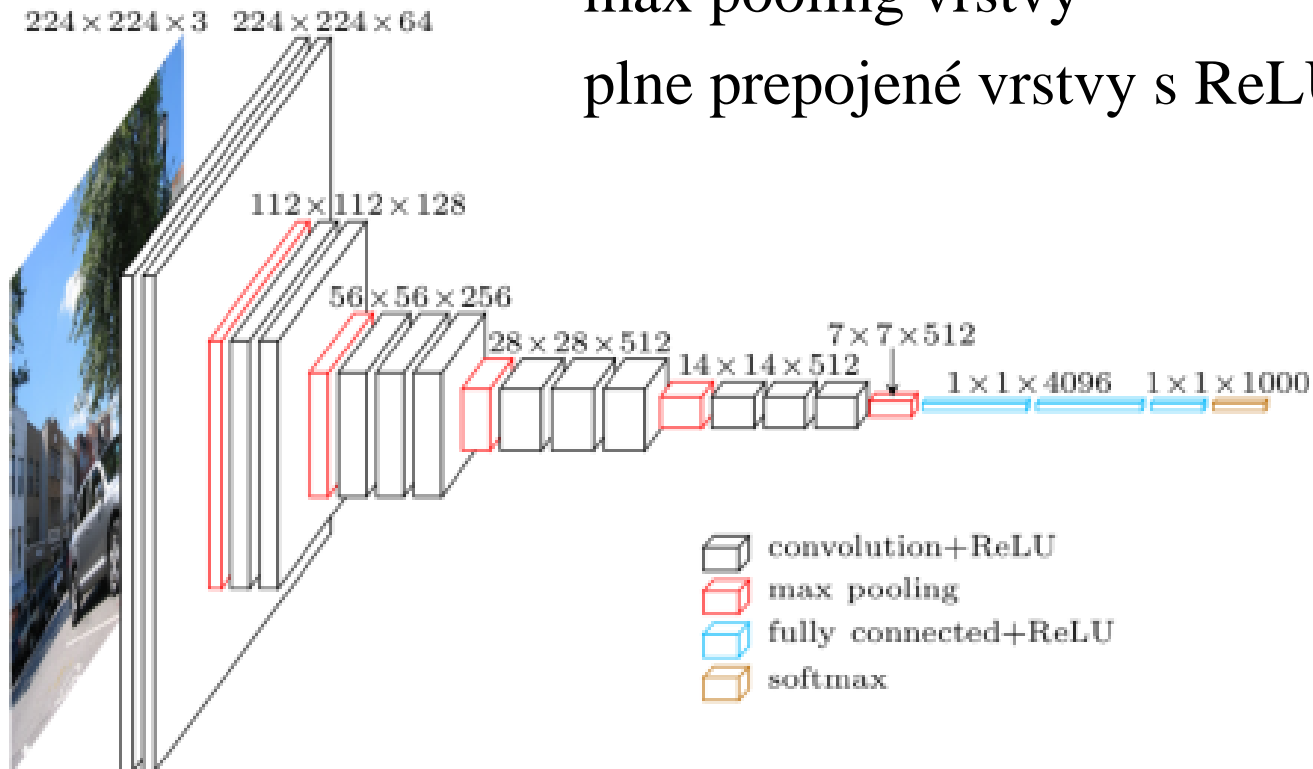
Každý neurón
výstupnej vrstvy má
spojenia na $3 \times 3 = 9$
neurónov vstupnej
vrstvy

Každý neurón má na
spojeniach rovnaké
váhy, tj. sieť má 9
parametrov

Sieť môže mať ako
10. parameter bias,
ktorý pri ReLU
aktivácii predstavuje
prah

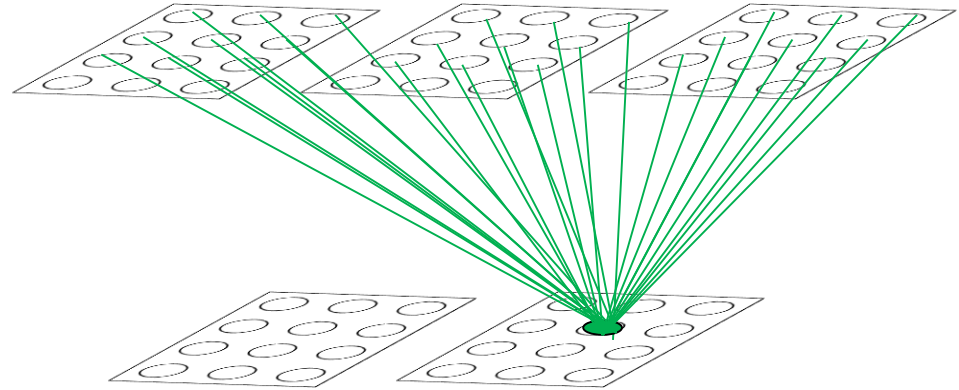
VGG

- konkrétna architektúra DNN navrhnutá pôvodne pre ImageNet, t.j. kategorizáciu obrázkov
- Používa 3 stavebné prvky: konvolučné vrstvy s ReLU aktiváciou
max pooling vrstvy
plne prepojené vrstvy s ReLU a Softmax



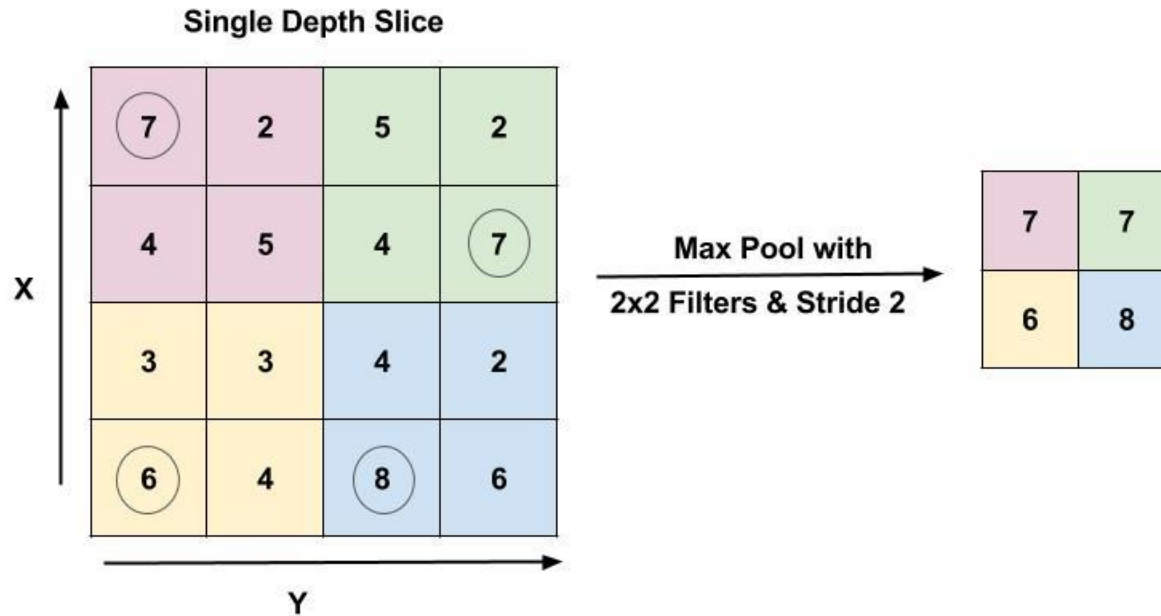
Conv2D Layer (CNN sieť)

- Konvolučná vrstva obsahuje jednu alebo viacero konvolučných sietí



- Všetky neuróny konvolučnej siete sú rovnaké t.j. majú rovnaké váhy a bias
- Každý neurón je spojený do všetkých batch častí vstupu kde na svoje miesto v dátovej zložke aplikuje kernel danej veľkosti

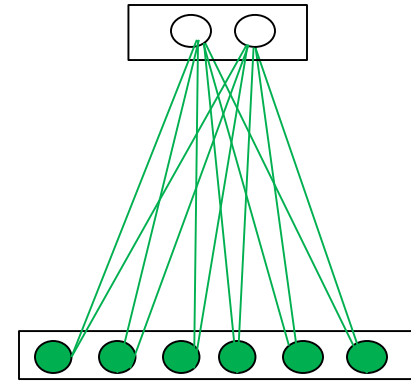
MaxPooling2D Layer



- Redukcia dát, aplikuje sa často po Conv2D
- $\text{pool_size} = (\text{sx}, \text{sy})$, $\text{strides} = \text{pool_size}$ (krok)

Dense Layer

- Vrstva neurónov rovnakého charakteru ako predošlá.
- Neuróny sú prepojené každý s každým
- Každý má svoje váhy a prípadne aj bias a aktivačnú funkciu

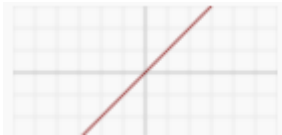


*Bias sa odpočíta od výstupnej hodnoty
Pri Dense layeri ho má každý neurón rôzny*

Aktivácia

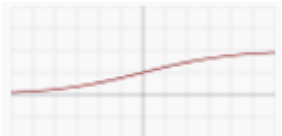
- Na výstup z neurónu je možné aplikovať aktivačnú funkciu

linear



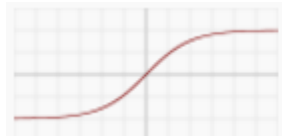
$$f(x) = x$$

sigmoid



$$f(x) = \frac{1}{1 + e^{-x}}$$

tanh



$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

relu



$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

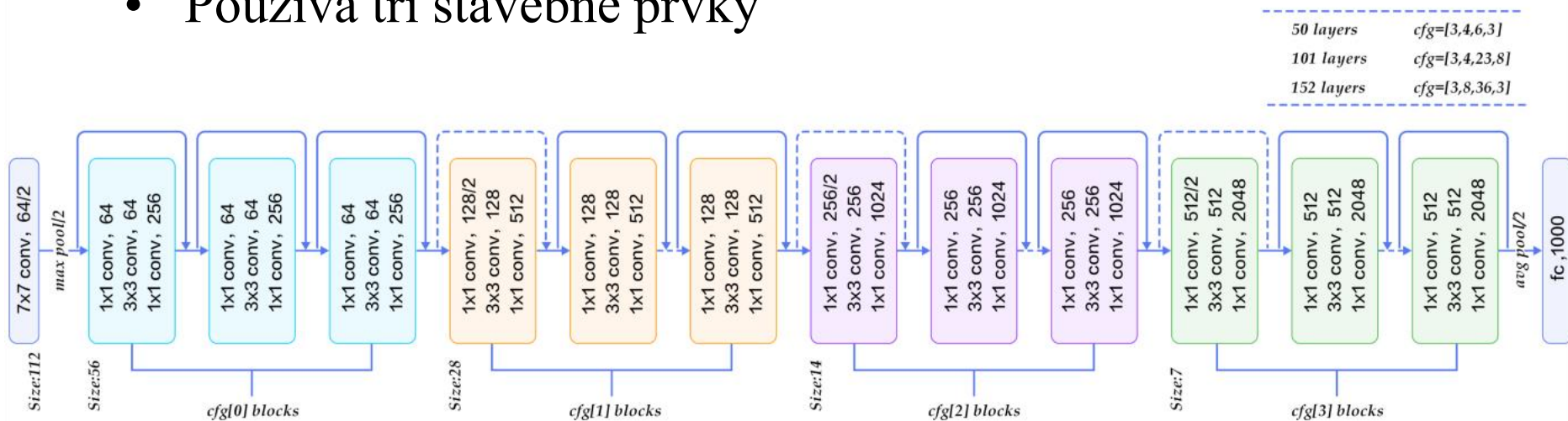
softmax

$$\begin{bmatrix} 1.2 \\ 0.9 \\ 0.4 \end{bmatrix} \xrightarrow{\text{Softmax}} \begin{bmatrix} 0.46 \\ 0.34 \\ 0.20 \end{bmatrix}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

ResNet

- konkrétna architektúra DNN navrhnutá pôvodne pre ImageNet, t.j. kategorizáciu obrázkov
- Vznikla ako vylepšenie VGG pridaním tzv. reziduálov, ktoré majú zabezpečovať riešenie problému „miznúcich dát (vanishing data)“
- Používa tri stavebné prvky



Trénovanie

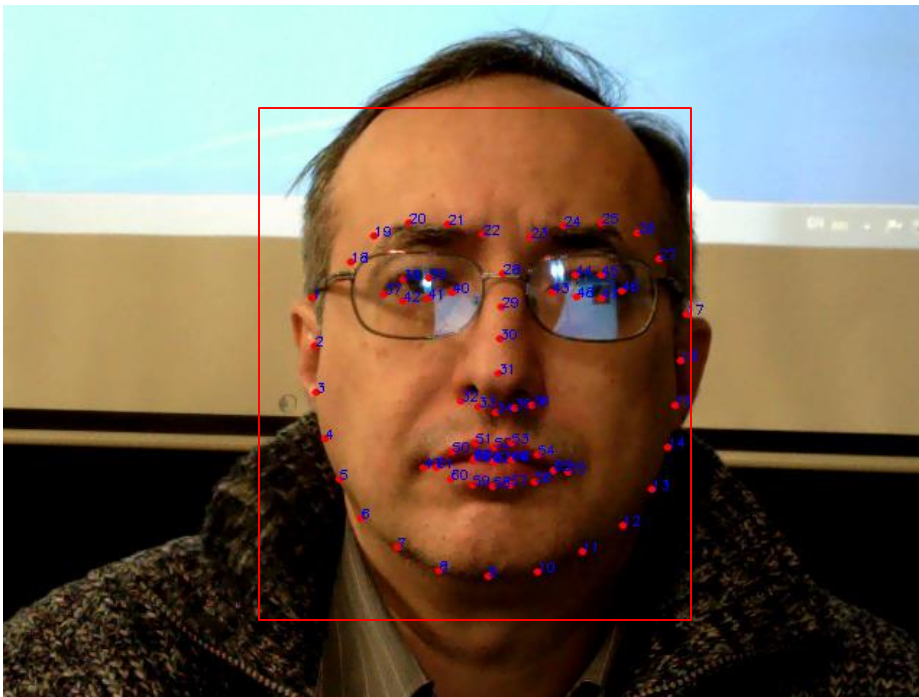
- Chybová funkcia: mse, metric
- Gradient descent
- Stratégia z koľkých vzoriek vyrátať gradient: Mini-batch gradient descent
- Stratégia ako odhadnúť skutočnú veľkosť gradientu: rmsprop, ADAM
- Stratégia ako odolávať preučeniu: dropout, batch normalization, xavier initialization
- Trénovať nemusíme, ak máme model a ani nemôžeme, ak nemáme dataset

Detekcia tváre

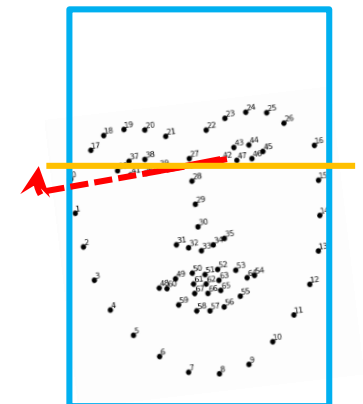
- OpenCV dodáva model na detekciu tváre `res10_300x300_ssd_iter_140000` ktorý dosahuje oveľa lepšie výsledky ako HOG detektor
- Na rozdiel od HOG spracúva celý obrázok a na výstupe dá zoznam vierohodností a obdĺžnikov, v ktorých sú na obrázky tváre
- Je to Caffe model, čiže je dodaný v podobe `.prototxt` (architektúra) a `.caffemodel` (váhy)

Črty tváre

- Nie je k dispozícii model, takže použijeme z klasického riešenia kaskádny regresor z knižnice Dlib



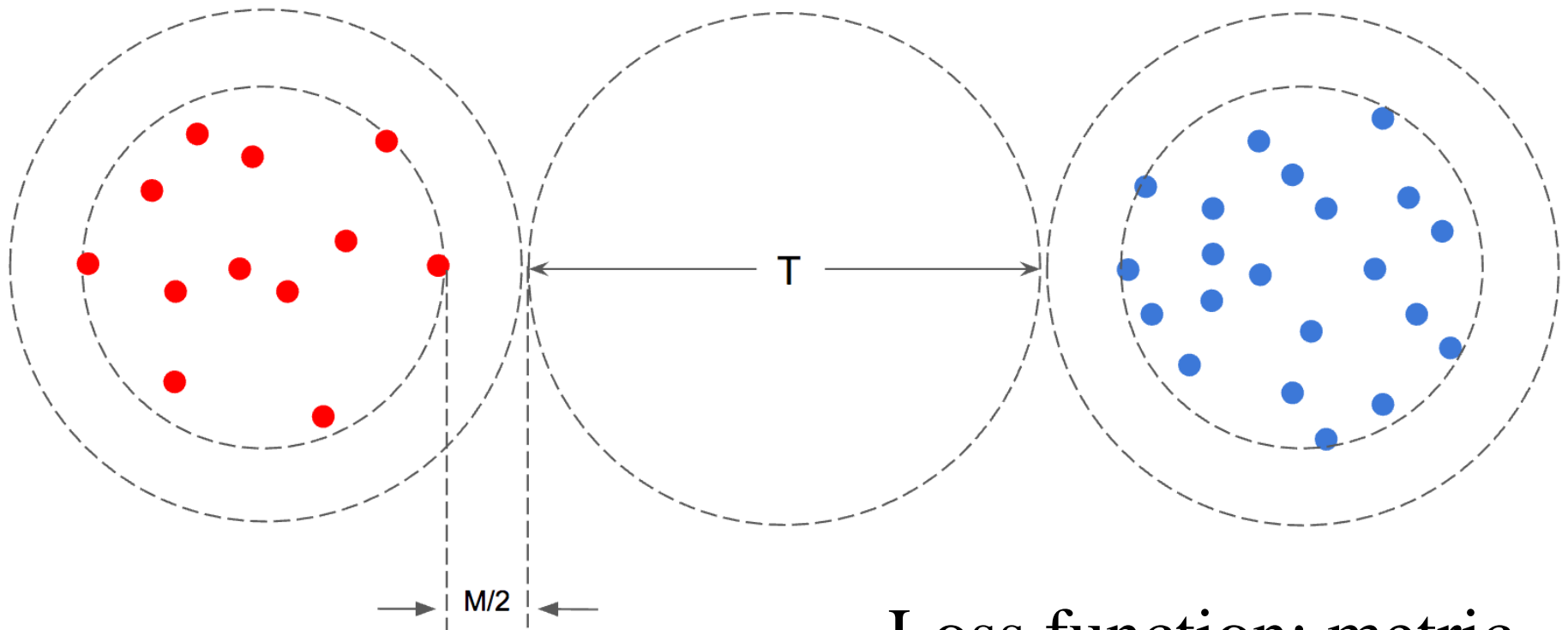
- Črty využijeme na transformovanie tváre do normalizovaného tvaru



Rozpoznanie tváre

- Dlib dodáva model `dlib_face_recognition_resnet_model_v1` na získanie descriptora tváre: DNN dostane tvár a vráti 128 floatov
- Pre tváre toho istého človeka majú ich descriptory k sebe blízko, pre tváre rôznych ľudí sú ďaleko od seba, v priestore $\langle 0, 1 \rangle^{128}$
- Ako každý Dlib model je dodaný v binárnej forme ako `.dat`
- Táto DNN je trénovaná dost' špeciálnym spôsobom

- Zvolia sa dve konštanty T a M
- V jednej epoche trénovania sa náhodne vyberie mini batch, v ktorom sú vzorky ktoré patria do viacerých kategórii
- Pre každú sa spustí sieť a tieto vzorky sa namapujú do priestoru deskriptorov
- Pre každú dvojicu patriacu k sebe sa otestuje, či sú bližšie ako $T-M/2$
- Pre každú dvojicu nepatriacu k sebe sa otestuje, či sú ďalej ako $T+M/2$
- Ak nie, tak sú zlé a dá sa pre každú zvlášť určiť ako sa má daný deskriptor zmeniť
- 15 najvýraznejších zmien sa postupne aplikuje ako gradient na úpravu váh siete



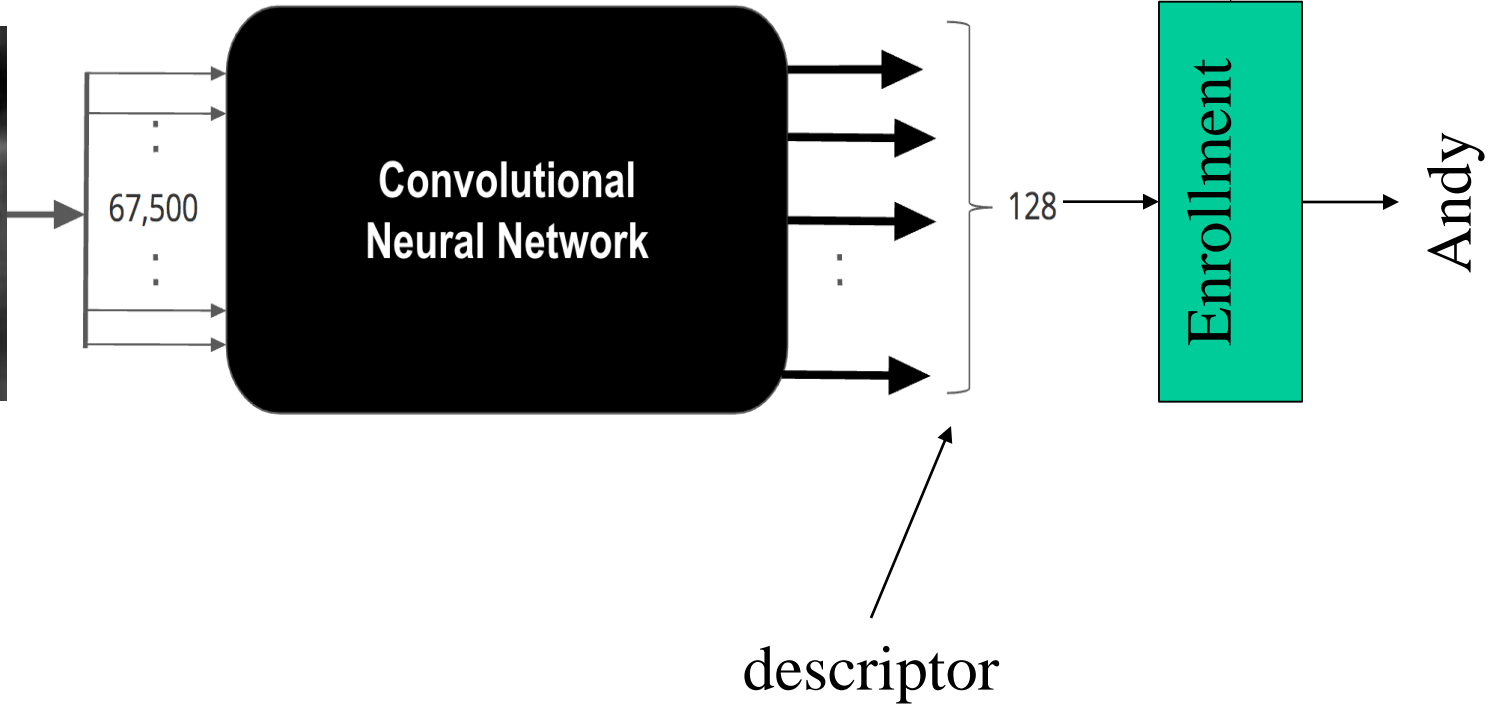
Loss function: metric

Enrollment („Listina přítomných“)

Určíme nejbližší descriptor ze vzorových příkladů a za toho člověka viděné vyhlásíme



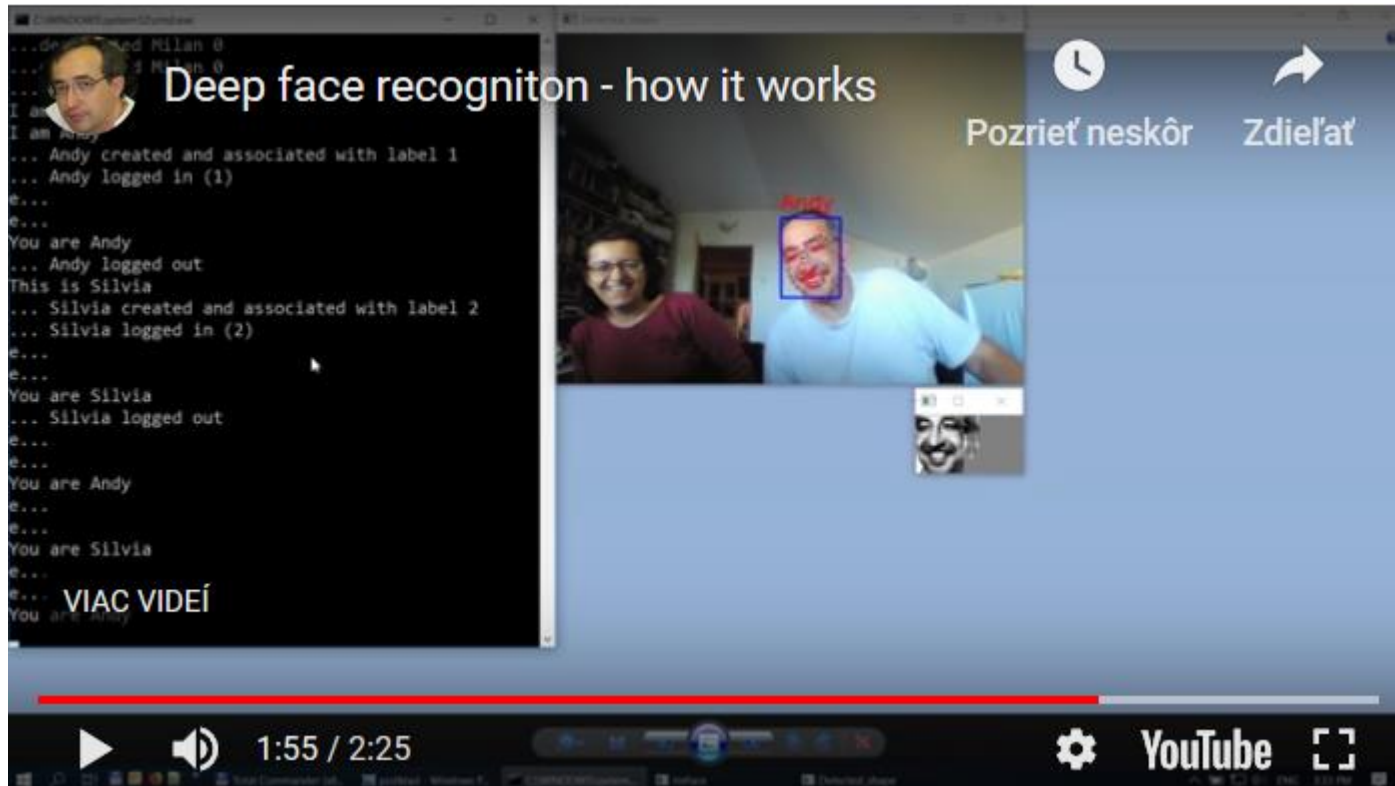
300x300



Interaktívne rozpoznávanie tváre

- Systém si spravuje databázu príkladov tvári určitých ľudí
- Niekoho dáme do centra obrazu a povieme ako sa volá
- Systém vždy vyhodnotí, kto to podľa neho je
- Ak je to niekto iný, ako sme mu povedali, tak si zavedie ďalší príklad do databázy
- Po nejakom čase sa systém saturuje (alebo zdegraduje – väčšinou preto, že si osvojil nejaký nevhodný vzor, ktorý v skutočnosti nebol tvárou)

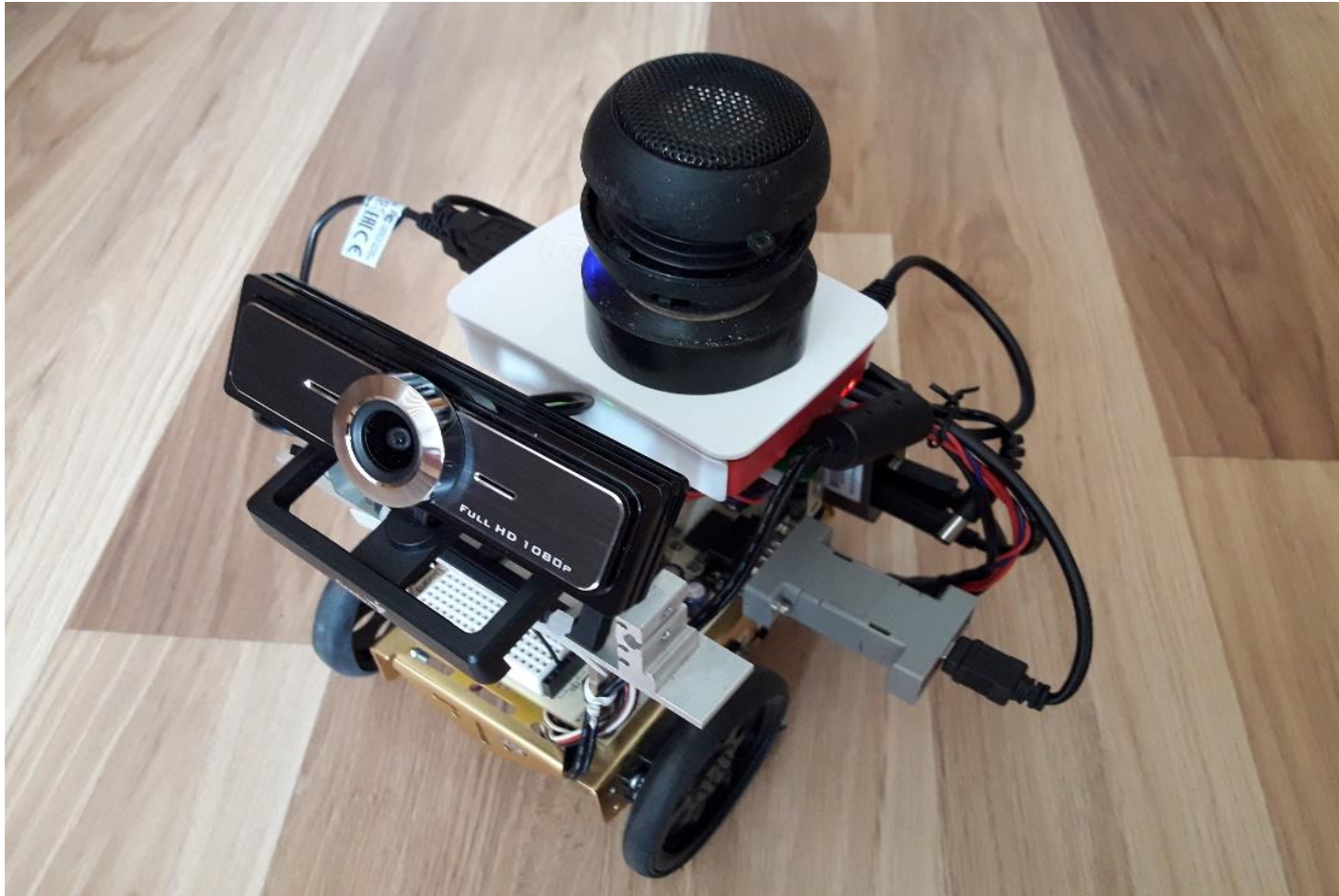
Testovanie na PC



https://youtu.be/IV3kuQnoC_E

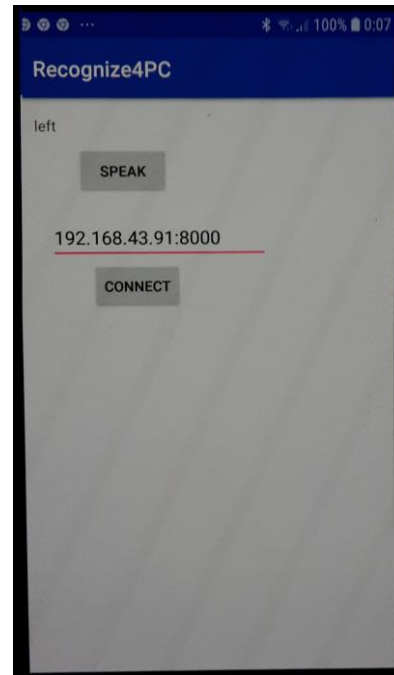
<https://youtu.be/bN5PUmpLsl4>

Testovanie na Robotovi



Ovládanie robota hlasom

- Ako robotovi povedať, koho vidí? Ideálne riešenie je povedať mu to.



<https://github.com/andylucny/Recognize4PC>

Ďakujeme za pozornosť

Seminár Robotika.SK

Rozpoznávanie tváre pomocou hlbokého učenia

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny