

Detekce kašle pomocí metod analýzy obrazu

Dušan Kučeřík, KUC0277

11. května 2020

Obsah

1	Úvod	2
2	Získání a příprava videa	3
2.1	Nahrání videa	3
2.2	Úprava videa pro knihovnu OpenPose	3
3	Knihovna OpenPose	3
4	Příprava dat	4
4.1	Rozdělení obličejových bodů	4
4.2	Normalizace dat	4
4.3	Označení jednotlivých snímků	6
5	Trénování a predikce dat pomocí LSTM neuronové sítě	6
5.1	Krátce o LSTM síti	6
5.2	Samotné trénování	6
6	Závěr	10

1 Úvod

Cílem projektu je pomocí metod analýzy obrazu a strojového učení rozpoznat kašlající osobu. Projekt se tedy bude zabývat nahráním a přípravou videa a následné detekcí obličejových bodů pomocí knihovny OpenPose, které budou využity pro rozpoznání, zda-li osoba kašle či provádí normální aktivitu, jako je například otáčení hlavy pro sledování provozu nebo komunikace s ostatními pasažéry.

2 Získání a příprava videa

Tato kapitola se věnuje popisu získání a přípravě videa, ať už jde o získání videa nahrávajícím zařízením nebo o následném nastříhání videa na jednotlivé snímky.

2.1 Nahrání videa

Nahrávání velké většiny testovacích videí probíhalo na osobní telefon iPhone 7 s rozlišením 1080x1920 a 30 snímky za vteřinu. Vyzkoušena byla i verze natáčení testovacích videí pomocí GoPro modelu HD z roku 2014. Ovšem kvalita natočeného videa z tohoto zařízení nedosahovala požadovaných kvalit pro správnou detekci tváře, takže pro nahrávání videí byla využita první verze. Nahrávání probíhalo ze dvou úhlů pohledu.

1. Pohled zepředu, kdy bylo nahrávací zařízení umístěno na čelním skle před řidičem ve výšce jeho hlavy.
2. Pohled z boku, kdy bylo nahrávací zařízení umístěno na čelním skle před místem spolujezdce. Tedy nahrávací zařízení bylo pootočeno o 45 stupňů směrem k řidiči.

2.2 Úprava videa pro knihovnu OpenPose

Nahraná videa poté byla upravena do požadovaného formátu pomocí softwaru pro úpravu video-sequencí. Video bylo oříznuto do rozlišení 800x800 obrazových bodů. Polohu obličeje jsem vycentroval do středové polohy a dále upravil světelné vlastnosti videa a doostřil. Video jsem následně nastříhal na jednotlivé snímky, na kterých jsem později mohl generovat obličejové body pomocí knihovny OpenPose.

3 Knihovna OpenPose

OpenPose představuje první systém pro detekci bodů lidského těla, rukou, obličeje a nohou (celkem 135 bodů) na jednotlivých obrazech. Dokáže v reálném čase detekovat buď 2D detekci bodů pro jednu či více osob z jednoho zdroje nahrávání, či 3D detekci bodů pro jednu či více osob z více zdrojů nahrávání. Výkon knihovny závisí hlavně na počtu osob v daném záběru. [1]

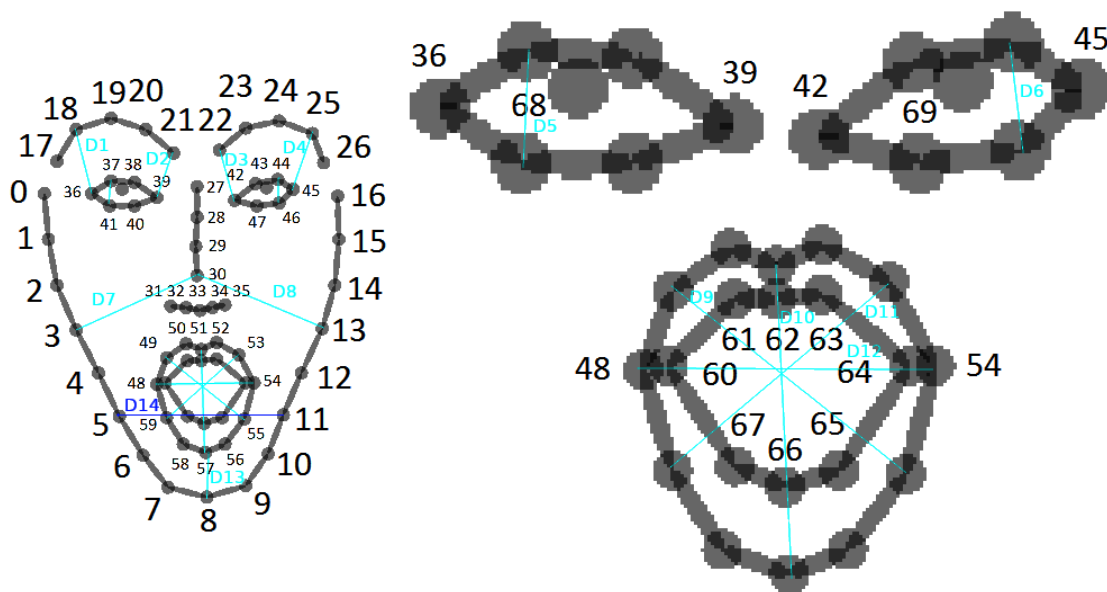
Pomocí knihovny OpenPose jsem vygeneroval obličejové data pro jednotlivé snímky z předem připraveného videa. Každý obličejový bod, který OpenPose generuje se skládá ze tří pozic a jeden obličej na každém snímku má 70 bodů. Tedy pro každý snímek OpenPose generuje 210 čísel. Jednotlivé obličejové body jsem vyexportoval do .CSV souboru, který později bude použitý pro rozpoznání kašlající osoby díky těmto obličejovým bodům.

4 Příprava dat

V této kapitole je popsáno, jak byla data získána z knihovny OpenPose připravena pro další použití v LSTM neuronových sítích nebo vizualizaci dat.

4.1 Rozdělení obličejových bodů

Po získání jednotlivých obličejových bodů pomocí OpenPose knihovny a načtení .CSV souboru s body jsem rozdělil (dle mého uvážení) vzdálenosti mnou vybraných obličejových bodů, které jsou znázorněny na Obrázku č.1. Jedná se hlavně o body, u kterých je možné, že bude docházet ke změně vzdálenosti při kašlání, mluvení či normálním stavu. Taktéž tyto body nebudou mít vliv na pohyb hlavy do stran, jelikož vzdálenosti jednotlivých bodů budou stále zachovány. Tedy porovnávat polohy bodů by dle mého uvážení nebylo korektní.



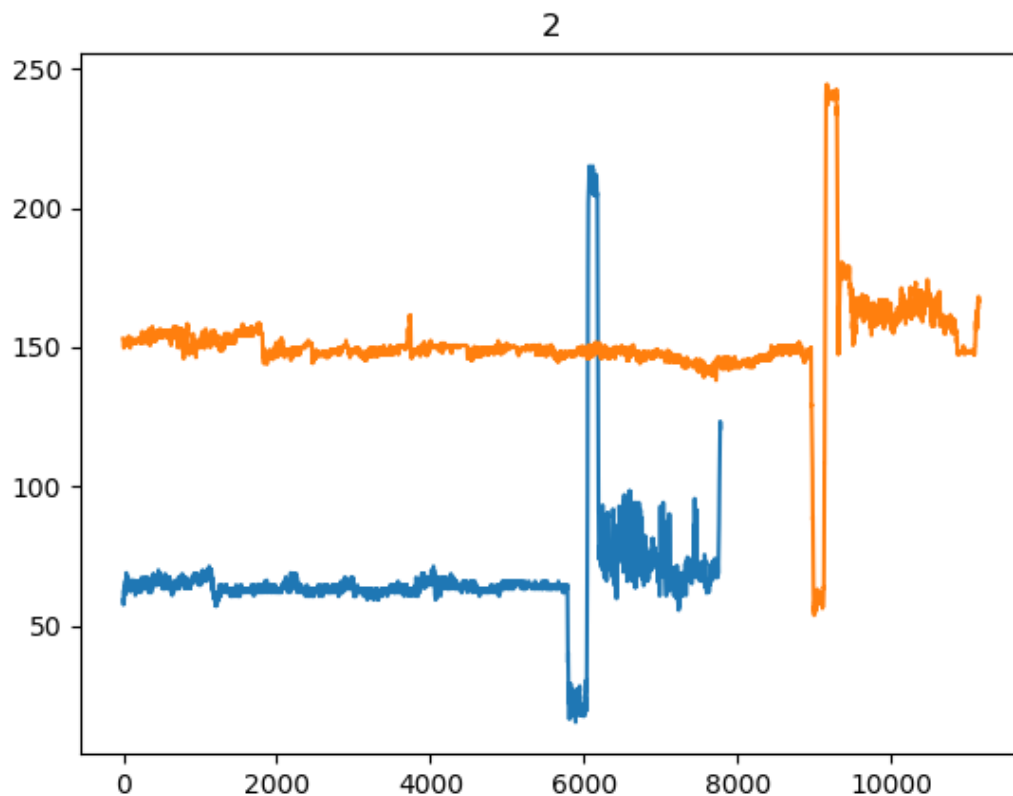
Obrázek 1: OpenPose obličejové body (vybrané vzdálenosti)

Tyto vzdálenosti D1-D14 jsem počítal pomocí Euklidovské vzdálenosti[3] dvou bodů A a B, kde výsledkem je číslo $ds(A, B)$ definováno vztahem

$$ds(A,B) = \sqrt{(a1 - b1)^2 + (a2 - b2)^2}$$

4.2 Normalizace dat

Zkusil jsem všechny data (jak data pro pohled zepředu, tak pohled zboku) normalizovat, aby se dal model strojového učení trénovat/testovat na spojených datech nezávisle na úhlu pohledu. Spojená data dvou zdrojů (zepředu, zboku) před normalizací vypadala následovně (modrá křivka: pohled zepředu, oranžová křivka: pohled zboku):



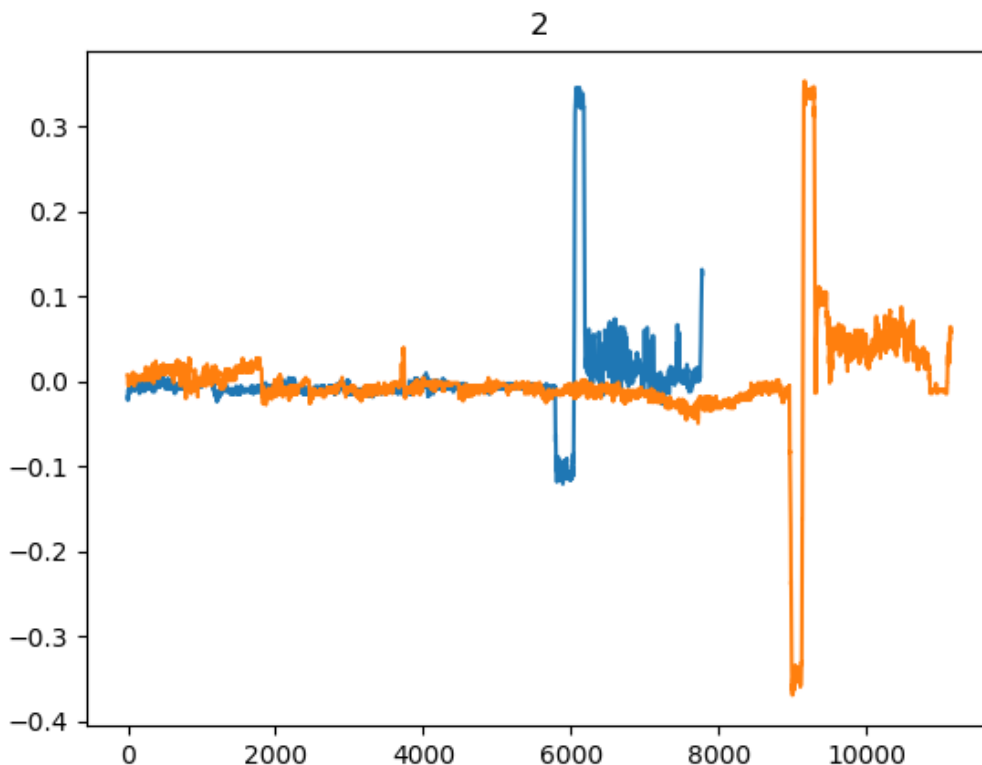
Obrázek 2: Spojená data dvou zdrojů před normalizací

Jednotlivé zdroje dat jsem tedy zkusil normalizovat pomocí Z-Score vzorce

$$\text{norm}(x) = (x - \text{mean}(\text{data})) / \text{var}(\text{data})$$

kde x je jeden bod ze zdroje dat data , $\text{mean}(\text{data})$ je průměr všech hodnot ze zdroje dat a var je rozptyl všech hodnot ze stejného zdroje dat.

Takto spojené data dvou zdrojů (zepředu, zboku) po normalizaci můžeme vidět na obrázku č.3 (modrá křivka: pohled zepředu, oranžová křivka: pohled z boku):



Obrázek 3: Spojená data dvou zdrojů po normalizaci

4.3 Označení jednotlivých snímků

Byla vyzkoušena i metoda AutoEncoderu (typ neuronové sítě bez dozoru), ovšem nebylo dosaženo uspokojivých výsledků. Nakonec jsem tedy přistoupil k typu neuronové sítě s dohledem nad trénovanými daty. To znamená, že jsem ručně jednotlivé snímky označil/oštkoval, zda-li se v daném snímku jedná o kašel či nikoliv.

5 Trénování a predikce dat pomocí LSTM neuronové sítě

5.1 Krátce o LSTM síti

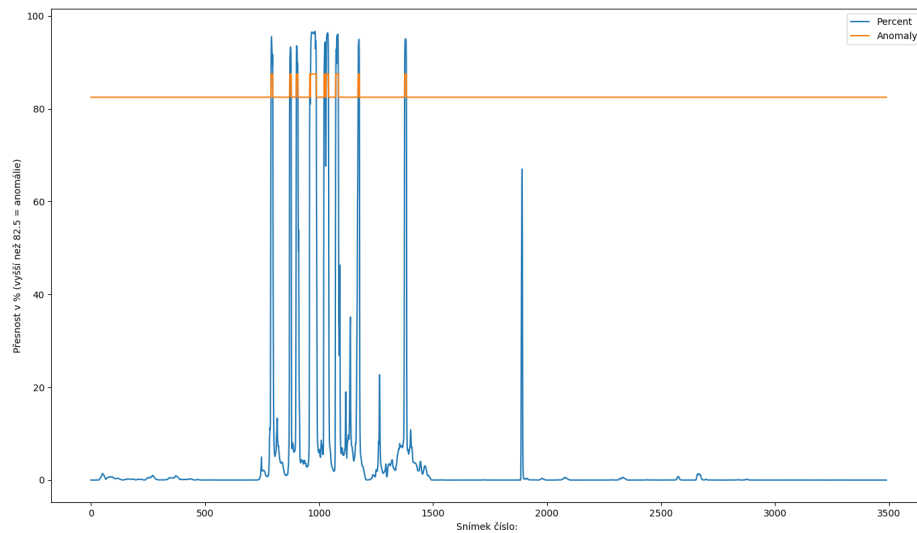
LSTM síť nebo-li taky Long Short Term Memory jsou neuronové sítě, které jsou schopné se učit dlouhodobé závislosti díky zpětným vazbám. Díky tomuto výsledek aktuálního výstupu závisí na předchozích vstupech. Díky tomu mají LSTM sítě uplatnění na velkou variaci problémů. [2]

5.2 Samotné trénování

Na trénovacích datech i na normalizovaných trénovacích datech bylo vyzkoušeno trénování modelu s různými parametry (menší/větší batch-size, počet epoch menší/větší learning rate, dropout rate nebo například lookback). Na testovacích datech poté byla ověřena správnost natrénovaného modelu.

Po normalizaci a spojení dat ze dvou zdrojů (pohled z boku, pohled zepředu) a následném natrénování na takových trénovacích datech, nebyly výsledky příliš přesné (jak lze vidět na obrázku číslo 4). Lze vidět, že model dokázal s přesností nad 80% odhadnout kašel, ovšem v intervalu, kde docházelo

ke kašlání, mnohokrát odhadoval kašel pouze s přesností okolo 10-20%. Po tomto zjištění bylo usouzeno, že trénovací/testovací data nemohou pocházet ze dvou zdrojů (pohled z boku, pohled zepředu) a následně být spojeny a normalizovány.

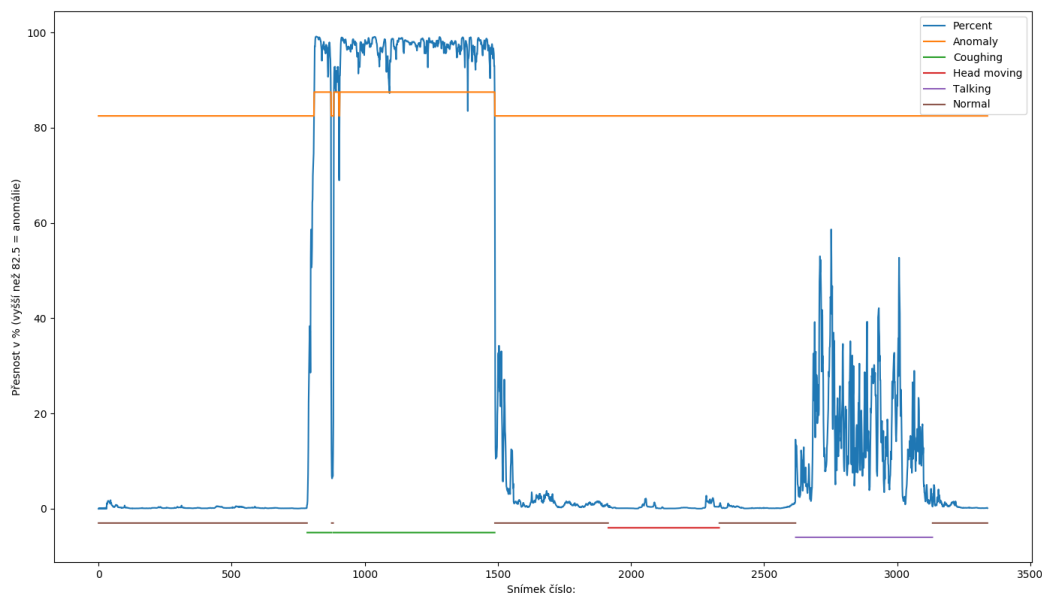


Obrázek 4: Testování natrénovaného modelu na normalizovaných datech ze dvou zdrojů

Ve výsledku byl tedy nejpřesnější model natrénovaný na ne-normalizovaných datech z jednoho zdroje dat (v tomto případě pohled zepředu) s následujícími parametry:

1. Lookback = 30 (Udává kolik snímků zpět model sleduje výsledky, podle kterých následně predikuje současný výsledek snímku)
2. Batch size = 128 (Definuje počet vzorků, které budou propagovány do sítě)
3. Dropout rate = 0.1 (Definuje kolik % vzorků bude vyloučeno z každého aktualizacího cyklu. Zlepšují výkon neuronové sítě a zabraňují přeplnění modelu.)
4. Epoch count = 35 (Udává počet trénovacích cyklů)
5. Learning rate = 0.0001 (Rychlost učení určuje, jak rychle je model přizpůsoben problému)

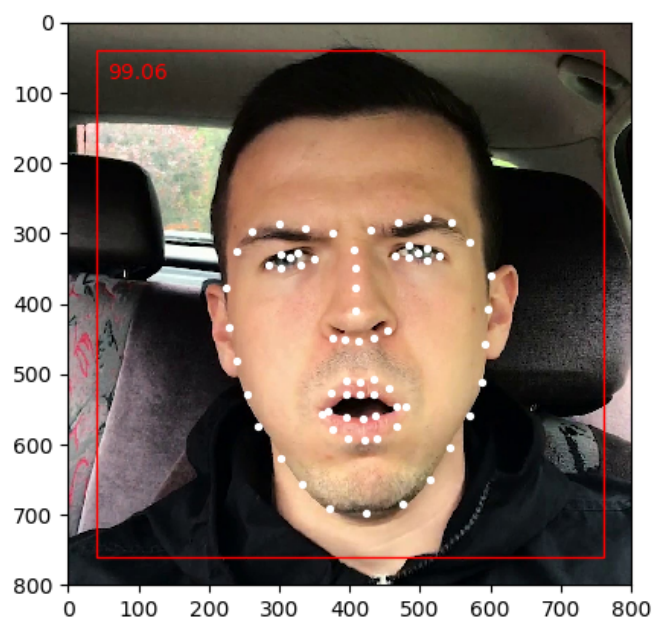
Testování takto natrénovaného modelu na ne-normalizovaných datech lze vidět na obrázku číslo 5.



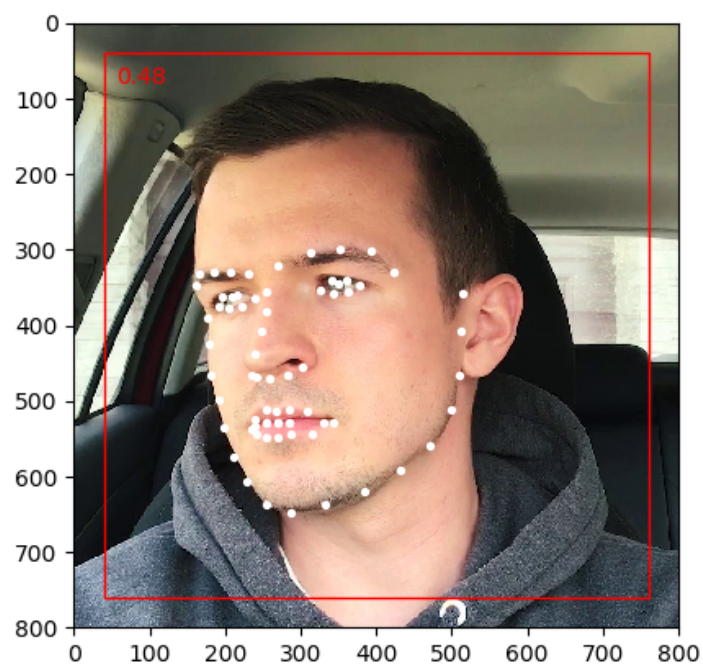
Obrázek 5: Testování natrénovaného modelu na ne-normalizovaných datech z jednoho zdroje

Na obrázku číslo 5 lze pozorovat různé chování testovaného subjektu. Testovací data obsahují normální chování řidiče, jež se skládá z otáčení hlavy pro sledování provozu a komunikace s ostatními pasažéry. Interval okolo snímku číslo 750 až 1500 se skládá z kašláním řidiče. Práh úspěšnosti odhadu kašláním byl nastaven na 82,5% a z obrázku číslo 5 lze tedy usoudit, že natrénovaný model s velkou přesností nad daný práh úspěšnosti odhadl velkou většinu snímků správně. Také lze pozorovat jistý nárůst odhadu kašláním při komunikaci subjektu, ale odhad se pohybuje okolo 20 až 30% a tudíž nepřesahuje práh úspěšnosti odhadu kašláním.

Na následujících obrázcích (č. 6 a č. 7) vidíme dva vybrané případy. Na prvním snímku se jedná o kašel a na druhém snímku o normální chování. V levém horním rohu pozorujeme odhad v %, který udává na kolik % model odhaduje, že v daném snímku testovaný subjekt kašle.



Obrázek 6: Vizualizace natrénovaného modelu - kašel



Obrázek 7: Vizualizace natrénovaného modelu - normální chování

6 Závěr

Cílem projektu bylo analyzovat obraz a pomocí strojového učení rozpoznat kašlající osobu. Tohoto cíle se z velké části podařilo dosáhnout. Proběhlo nahrání jedné desítky testovacích/trénovacích videí pro neuronovou síť, které se podařilo úspěšně upravit do jednotné podoby pro knihovnu OpenPose, aby mohla úspěšně generovat obličejové body zhruba ve stejném datovém rozmezí. Dále byla vyzkoušena varianta se spojením dvou zdrojů dat, respektive dat z dvou různých úhlů pohledu. Ovšem tato varianta po testovacích pokusech nebyla tak přesná, jak se očekávalo. Nakonec tedy byla použita varianta pouze pro trénování a testování pouze z jednoho úhlu pohledu. Po měřených výsledcích můžeme říct, že takto natrénovaný model dokázal z velké části rozpoznat kašlající osobu, či osobu v normálním stavu (mluvení, otáčení hlavy).

Reference

- [1] CMU-Perceptual-Computing-Lab: OpenPose. Github.com [online]. [cit. 2020-05-11]. Dostupné z: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>
- [2] Rekurentní neuronové sítě pro klasifikaci textů. vutbr.cz [online]. [cit. 2020-05-11]. Dostupné z: https://www.vutbr.cz/www.base/zav_prace_soubor_verejne.php?file_id=171408
- [3] Three-dimensional Coordinate Systems. Math.usm.edu [online]. [cit. 2020-05-11]. Dostupné z: <https://www.math.usm.edu/lambers/mat169/fall09/lecture17.pdf>
- [4] Grammatical Facial Expression Recognition. semanticscholar.org [online]. [cit. 2020-05-11]. Dostupné z: <https://pdfs.semanticscholar.org/815a/9f6f233815dca50fe6c99e5d7127ee43f3f3.pdf>
- [5] Step-by-step understanding LSTM autoencoder. towardsdatascience.com [online]. [cit. 2020-05-11]. Dostupné z: <https://towardsdatascience.com/step-by-step-understanding-lstm-autoencoder-layers-ffab055b6352>
- [6] Deep Learning For Hackers. github.com [online]. [cit. 2020-05-11]. Dostupné z: <https://github.com/curiously/Deep-Learning-For-Hackers>
- [7] Time Series Classification for Human Activity Recognition with LSTM. curiosily.com [online]. [cit. 2020-05-11]. Dostupné z: <https://www.curiously.com/posts/time-series-classification-for-human-activity-recognition-with-lstms-in-keras/evaluation>
- [8] Understanding LSTM networks. colah.github.io [online]. [cit. 2020-05-11]. Dostupné z: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [9] Bidirectional LSTM network. machinelearningmastery.com [online]. [cit. 2020-05-11]. Dostupné z: <https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras/>