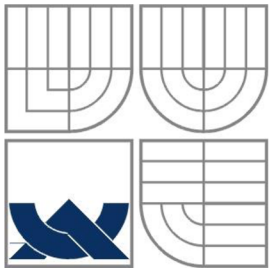
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

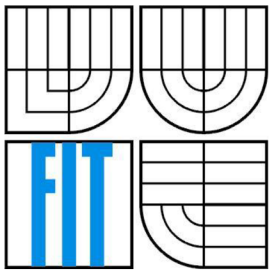
Fakulta informačních technologií
Faculty of Information Technology

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

Brno, 2016 JAN JILEČEK



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ROZPOZNÁVÁNÍ EMOCÍ POMOCÍ KONVOLUČNÍCH NEURONOVÝCH SÍTÍ

CONVOLUTIONAL NEURAL NETWORKS FOR EMOTION RECOGNITION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAN JILEČEK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL HRADIŠ, Ph.D.

BRNO 2016

Abstrakt

Konvoluční neuronové sítě se dnes používají v mnoha oblastech, především ale pro strojové učení, kde vykazují velkou úspěšnost. Tato práce nejprve představí existující frameworky, další algoritmy pro rozpoznávání a pak popisuje, jak probíhalo vytváření vlastní datové sady a trénink modelu pro rozpoznávání emocí. Tento model má úspěšnost klasifikace 60%. Model je následně využit pro získání statistik o emocích z filmových trailerů a z těchto statistik je sestaven model pro rozpoznávání žánrů, který je konečně použit v naší aplikaci pro určení žánru vstupního traileru s přesností až 47%.

Abstract

Convolutional neural networks are used for various tasks, but foremost in machine learning, in which they excel. This work is going to introduce some existing frameworks, other algorithms for recognition and then we describe the training dataset creation and the model for emotion recognition training process. Mentioned model has accuracy of 60%. It is used for emotion statistics retrieval from movie trailers. Model for genre recognition is created from those statistics and then finally used in our application for genre recognition of the input trailer, with best accuracy of 47%.

Klíčová slova

Konvoluční neuronové sítě, Rozpoznávání obličejů, Rozpoznávání emocí, Rozpoznávání žánrů, Caffe framework, CK, AMFED, KDEF, SFEW, Brazilian FEI, Algoritmus K-nejbližších sousedů, OpenCV

Keywords

Convolutional neural networks, Facial recognition, Emotion recognition, Movie genre recognition, Caffe framework, CK, AMFED, KDEF, SFEW, Brazilian FEI, K-nearest neigbours, OpenCV.

Citace

Jileček Jan: Rozpoznávání emocí pomocí konvolučních neuronových sítí, bakalářská práce, Brno, FIT VUT v Brně, 2016

Rozpoznávání emocí pomocí konvolučních neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, Ph.D.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Jileček
17. 5. 2016

Poděkování

Tímto bych rád poděkoval vedoucímu mojí bakalářské práce, panu Ing. Michalu Hradišovi, Ph.D., za jeho skvělé vedení a pomoc. Dále bych chtěl poděkovat za možnost využití služby MetaCentrum, poděkování je ve vyžadovaném formátu uvedeno níže.

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

Obsah

Seznam obrázků

1 Úvod

Konvoluční neuronové sítě se v dnešní době používají v mnoha různých oblastech, především jsou ale použity při zpracování obrazu. V této práci jsou užity při klasifikaci emocí z obličejů přítomných ve snímcích videa, k tomuto účelu je použit open-source framework Caffe[1]. Hlavní řešený problém je určování žánru videa pomocí emocí přítomných ve videu. Nejprve bylo třeba nashromáždit data, na kterých byla následně natrénována neuronová síť. Dále byla provedena detekce tváří a určování emocí tváře na snímcích získaných z celkem 593 filmových trailerů, kde byl každý ze šesti žánrů zastoupen přibližně 100 trailery. Testovací sada je tvořena dalšími 500 trailery. Pro určování podobnosti podle informací o emocích ve videu jsem vytvořil vlastní algoritmus, experimenty byly prováděny také se Support Vector Machines, dále SVM a algoritmem k-nejbližších sousedů, dále K-NN.

Po rozsáhlém automatickém i manuálním testování a pokusech jsem vybral síť s nejlepší přesností. Síť je natrénovaná na stažených i manuálně vytvořených datových sadách. Tato síť je hlavním stavebním kamenem našeho programu, a je použita pro klasifikaci veškerých emocí a byla užita jak pro trénování, tak pro testování. Úspěšnost klasifikace s touto sítí je 60%. Dalším krokem bylo využití této sítě pro vytvoření statistik o emocích získaných z trailerů jednotlivých žánrů. Nejlepší přesnosti pro určování žánru traileru prostřednictvím zjišťování emocí potom dosáhl algoritmus K-NN, a to 47% pro správné určení žánru na první pozici; přesnost určování až do třetí pozice byla testována pomocí vlastního algoritmu, v tomto případě pak bylo dosaženo přesnosti 86%.

Veškeré trailery byly staženy ze serverů Youtube.com[2], informace o žánrech filmů a jejich popularitě byly získány z mezinárodní filmové databáze imdb.com[3] (trailery filmů pro jednotlivé žánry pro trénování a testování nebyly stahovány náhodně, ale podle žebříčků populárnosti).

V druhé kapitole budeme probírat základy konvolučních neuronových sítí a charakterizujeme jednotlivé vrstvy a jejich funkci. Kapitola tři se věnuje námi používaným frameworkem Caffe, konkrétně základy jeho architektury, práce s ním a popíšeme si způsob jeho konfigurace.

Další kapitola se dedikuje porovnání současných frameworků pro strojové učení a práci s konvolučními neuronovými sítěmi. Kromě toho si také uvedeme algoritmy pro rozpoznávání tváří, spolu s příklady.

V šesté kapitole se budeme věnovat možnostem rozpoznávání emocí, a popíšeme si způsoby rozpoznávání z fyziologických vlastností člověka, hlasu a psaného textu.

Jako další si přiblížíme způsob vytváření trénovací datové sady pro naší neuronovou síť, problémy s tím spojené a způsob experimentace a hledání nejlepších parametrů. Následovat bude popis přípravy dat pro učení se z trailerů a ihned potom budou detailně popsány použité algoritmy pro trénování a klasifikaci emocí z trailerů. V poslední kapitole si popíšeme výslednou aplikaci.

[1] http://caffe.berkeleyvision.org/
[2] https://www.youtube.com/
[3] http://www.imdb.com/

2 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou speciálním typem neuronových sítí. Byly navrženy zejména pro rozpoznávání dvourozměrných obrazových dat, obrázků. Díky konvoluci není síť tolik náchylná k chybám, jelikož konvoluce do jisté míry ignoruje posuny nebo jiné deformace vstupních dat. V těchto sítích se používá algoritmus zpětného šíření chyby, backpropagation, který od konce předává předchozím vrstvám vypočítané chyby jednotlivých neuronů v závislosti na jejich vahách [1].

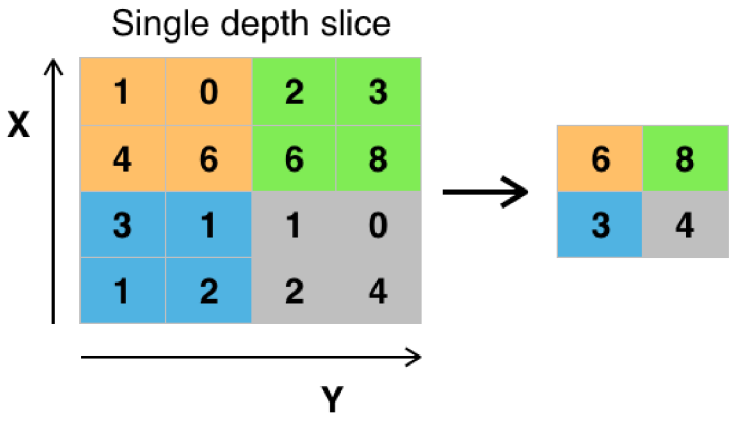
Konvoluční síť je tvořena, jak název napovídá, více konvolučními vrstvami. Mezi těmito vrstvami často bývá vrstva, která je zodpovědná za tzv. pooling, shlukování okolních pixelů do jednoho a tím postupné zmenšování velikosti vstupního obrázku. Po těchto vrstvách většinou následuje plně propojená vrstva, která propojuje všechny výstupy předchozí vrstvy se všemi svými vstupy. Mezi další používané vrstvy se řadí například Dropout, provádějící náhodné vynechávky dat pro lepší generalizaci, nebo Softmax, která na konci sítě určuje pravděpodobnosti jednotlivých určených tříd. V dalších odstavcích si popíšeme pouze první dvě hlavní.

2.1 Konvoluční vrstva

Vrstva obsahuje sadu filtrů, také zvaných kernel (jádro), které jsou zodpovědné za vytváření příznakových map. Tyto mapy jsou výsledkem konvoluce daného filtru přes celou šířku a výšku vstupního obrázku. Tyto filtry jsou použity pro celou šířku a výšku, aby byla umožněna správná detekce požadovaných vlastností nezávisle na posunutí obrázku. Výsledkem je skalární součin dat obrázku a filtru.

2.2 Sdružovací vrstva

Také tzv. pooling vrstva, má na starosti sdružování okolních pixelů do jednoho, čímž zmenšuje rozměry výstupních dat a tím i počet aktivací neuronů potřebných pro jejich zpracování. Běžně se používá jedna ze dvou hlavních verzí této vrstvy, a to s určováním maximální hodnoty anebo hodnoty průměrné, lze ale použít i například L2-norm [1]. Ke svojí práci používá parametru stride (krok), který určuje, po kolika pixelech se bude shlukovat. Běžná hodnota je 2, což na 2-rozměrném obrázku činí 2x2 hodnoty, tím pádem se zvolená funkce aplikovat na 4 hodnoty. Kvůli ztrátové povaze tohoto algoritmu může příliš velký parametr stride být nevhodný při potřebě zjišťovat i malé detaily, například u tváří. Vizualizaci práce algoritmu lze vidět na Obrázku 1. [2]



Obrázek 1- max pooling

3 Framework Caffe

3.1 Architektura frameworku Caffe

Caffe je framework pro strojové učení pomocí konvolučních neuronových sítí, který byl původně vyvinut na univerzitě Berkeley, nyní je udržován BVLC, centrem pro počítačové vidění v Berkeley.

Caffe poskytuje různé algoritmy pro strojové učení a také umožňuje snadné sdílení natrénovaných modelů nebo jejich stažení a následnou úpravu na vlastních datech. Podporuje CUDA architekturu pro rychlé výpočty na grafických kartách, umožňuje rovněž trénování na CPU. Je naprogramován v C++. Caffe díky svojí architektuře zvládne natrénovat síť na až 40 milionech obrázků za den při použití grafických karet jako jsou nVidia Titan nebo K40. [3]

Caffe umožňuje snadno přidávat nové typy vrstev a nové datové typy. Konfigurace je také velmi snadná díky podpoře jazyka Protocol Buffer. Při práci si Caffe rezervuje jen tolik paměti, kolik potřebuje. Caffe má také python a matlab rozhraní pro snadné a rychlé návrhy.

Caffe používá k ukládání informací mezi vrstvami takzvané bloby. Bloby jsou 4-dimenzionální datové struktury, které v sobě drží sady dat, jako například obrázky a dále obsahují parametry modelu. Při načítání dat do blobu je užito výkonu CPU a při práci s daty uvnitř je používána grafická karta.

Natrénované modely jsou na disk uloženy v již zmíněném formátu vytvořeném společností Google, Protocol Buffers. Umožňují snadnou serializaci dat a poskytují rozhraní jak pro Python, tak Matlab.

Vrstvy ve frameworku Caffe dostávají na vstup jeden nebo více již zmíněných blobů, a na výstupu vyprodukují další blob. Mezi vrstvami se postupuje buď dopředu, tzn. forward pass, nebo zpětně, tzv. backward pass. Forward pass vypočítává ze vstupu výstupy a backward passu je využit algoritmem zpětného šíření chyby pro promítnutí spočítaných vah na předchozí prvky sítě za účelem zmenšení výsledku loss funkce na co nejmenší hodnotu.

3.2 Konfigurace

Pro nastavení parametrů pro trénování nové sítě používá Caffe konfiguračních soubory obsahující definici všech vrstev, jejich parametry a vzájemné vztahy. Jako první je definována datová vrstva, u které je uvedeno umístění načítaných dat a velikost dávky (batch). Následuje konvoluční vrstva, která byla popsána dříve, stejně jako Pooling vrstva a Plně propojená vrstva. Mezi používané vrstvy ještě patří Dropout a Softmax vrstvy. Další potřebný soubor určuje parametry použité při učení sítě. Mezi tyto parametry patří cesta ke konfiguračnímu souboru s definicí vrstev, potom jak často se bude testovat, kolik položek bude načteno při testování, základní rychlost učení, politiku učení (inverse decay, krok, polynomická, sigmoidní a další) a parametry jako gamma, velikost kroku (při krokové politice), maximální počet iterací, momentum (zodpovědné za přidávání části váhy k současné váze) a ještě například jako často ukládat snímky sítě a nakonec jestli se bude síť počítat na procesoru nebo grafické kartě.

Při trénování používá Caffe již zmíněné algoritmy, včetně Backpropagation, algoritmus, který používá backward pass pro zpětnou úpravu parametrů. Jedná se o zpětné šíření chyby v závislosti na vahách jednotlivých neuronů, postupuje se podobně jako při forward pass, pouze v obráceném směru. Caffe při počítání tiskne informace o aktuální iteraci na standardní výstup. Nastavením parametrů je možné začít trénovat z již existující sítě a pouze jí takto upravovat k lepšímu, tzv. finetuning.

V našem projektu používáme pro vstupní data formát LMDB - Ligtning memory mapped database. Jedná se rychlou a paměťově nenáročnou databázi, jež používá mapování souborů do paměti, díky čemuž je extrémně rychlá a zároveň je stejně perzistentní jako běžné databáze s diskovými úložišti. [4]

4 Existující frameworky pro práci s neuronovými sítěmi

4.1 Caffe

Námi zvolený framework, popsaný v kapitole 3. Zvolen hlavně pro jeho jednoduchost a rychlost, dostupnost již natrénovaných modelů a rozhraní pro Python. Teď si porovnáme jeho výhody a nevýhody s ostatními frameworky.

4.2 Lasagne

Lasagne je knihovna pro práci s neuronovými sítěmi v Theanu, matematické knihovně pro Python. Stejně jako Caffe podporuje konvoluční neuronové sítě, dále podporuje rekurentní sítě, a to včetně LSTM, long short-term memory, která vyniká při předpovídání časových údajů natrénovaných na datech, kde mezi jednotlivými údaji bývá měnící se časový rozdíl. Lasagne je určeno zejména pro ty, kteří již mají zkušenosti se psaním matematických funkcí za použití knihovny Theano. [5]

4.3 Keras

Knihovna pro Python, která dokáže spolupracovat s již zmíněným Theanem a také s knihovnou TensorFlow, která používá pro výpočty grafy datových toků. Podporuje jak normální konvoluční neuronové sítě, tak rekurentní. Dokáže, stejně jako Caffe, bez povšimnutí přepínat mezi prací na CPU a GPU. [6]

4.4 OpenDeep

Stejně jako Caffe je tento Framework snadno rozšiřitelný a plně modulární. Jako předchozí uvedené frameworky používá rovněž knihovnu Theano. Navíc v sobě Framework obsahuje vestavěné nástroje pro vizualizaci a ladění trénovaných neuronových sítí. Přímo také spolupracuje s knihovnami jako numpy, scipy, pandas a scikit-learn. [7]

4.5 Torch7

Framework pro vědecké výpočty s podporou pro algoritmy pro strojové učení. Převážně zaměřen na práci na GPU, k dispozici je i podpora CUDA. Hlavním jazykem tohoto frameworku je Lua, přesněji používající kompilátor LuaJIT. Dalšími jazyky jsou C a C++. Hlavní výhodou Torch7 je nepřeberné množství volně dostupných komunitně vytvořených balíčků z různých oblastí, jako například počítačové vidění, strojové učení nebo zpracování signálů. Framework lze snadno spustit na jakékoliv platformě, díky implementaci v jazyce Lua. [8]

5 Způsoby rozpoznávání tváří

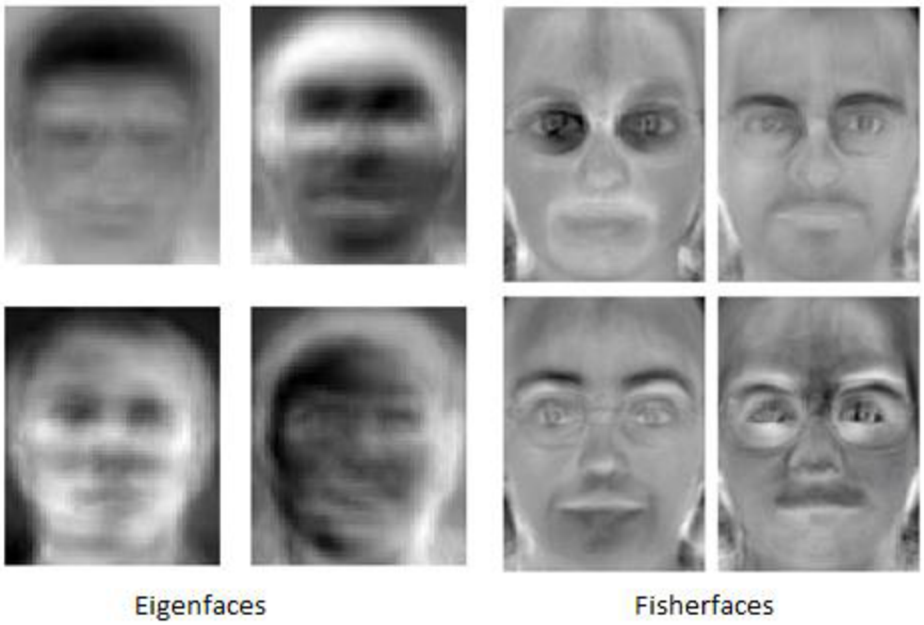
První algoritmy pro rozpoznávání tváří používaly detekci na základě bodů určujících rysy tváře. Další algoritmy již využívaly komplexnějších algoritmů, které budou v následujícím textu popsány. Mezi ně se řadí i rozpoznávání tváří pomocí neuronových sítí, čemuž se budeme věnovat v kapitole 9.

5.1 PCA a LDA (FLD)

PCA (Principal Components Analysis) převádí 2-rozměrný vektor, v našem případě obrázek, do 1-rozměrného vektoru. Jeho funkcí je zjištění takových vlastností obrázku, které se nejvíce liší od zbytku obrázku. Při tomto procesu je zahozeno velké množství dat. Je tak potřeba jen zlomek dat původního obrázku pro určování. Získaný obrázek je potom porovnáván s databází a je určována vzdálenost mezi naučenými vektory vlastností. Aby PCA fungovalo správně, musí být podobné osvětlení, velikost obrázku a pozice tváře. [9]

Dalším přístupem je LDA – Linear Diskriminant Analysis, statistický přístup založen na podobném způsobu jako PCA. Někdy se název LDA volně zaměňuje s názvem FLD, Fisher's Linear Diskriminant.

Výstup PCA je nazýván Eigenface, výstup FLD pak Fisherface. Příklady uvedeny na obrázku 2 níže: [10] [11]



Obrázek 2 - Eigenfaces a Fisherfaces

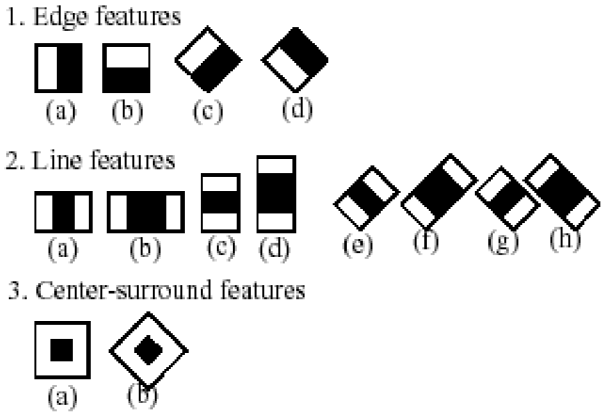
5.2 Kaskádové klasifikátory Haar vlastností

V našem projektu budeme používat kaskádové klasifikátory Haar vlastností za pomoci OpenCV knihovny pro Python.

Kaskádové klasifikátory jsou trénovány jak na položkách, jež obsahují objekt, který mají za úkol se naučit, tak na položkách, které daný objekt neobsahují. Všechny tyto obrázky jsou převedeny na společnou velikost. Po natrénování může být model použit pro klasifikaci, jejíž první fáze probíhá podobným způsobem, jako konvoluce – je vytvořeno okno, nebo by se také dalo říci výřez, kterým je postupně přejížděno po celé ploše obrázku. Nyní nastupují na scénu jednotlivé vrstvy klasifikátoru, tzv. kaskády. Testuje se totiž postupně od nejobecnější vrstvy a postupuje se dolů k detailnějším tak dlouho, dokud je výsledek detekce objektu pomocí určené vrstvy úspěšný. Pokud detekce úspěšná není, posunuje se okno detektoru dál po obrázku. Pro lepší detekci je vhodné měnit velikost „okna“, ve kterém se vyhledává, protože architektura kaskádových klasifikátorů umožňuje snadnou změnu velikosti klasifikátoru. [12]

V naší práci používáme soubor haarcascade_frontalface_alt.xml[4], jenž obsahuje informace o kaskádovém klasifikátoru pro detekci tváří zepředu. V našem projektu je klasifikátor nastaven tak, aby hledal tváře pouze v oblastech větších než 20x20px. Objekty by mohly být chybně klasifikovány na menší ploše.

Základní Haar vlastnosti jsou následující, obrázek 3 [13]:



Obrázek 3 - Haar vlastnosti

Příklad detekce pomocí kaskádových klasifikátorů za užití knihovny OpenCV je na obrázku níže, kde můžeme vidět jednoduchou detekci tváře. Pomocí OpenCV můžeme zvýraznit tvář, nebo ji oříznout. Ořez obrázku pro získání tváře je základní operace potřebná pro většinu úkolů v této práci. Obrázek 4 níže.

Obrázek 4 – Příklad detekce obličeje, Birdman

[4] https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalface_alt.xml

6 Způsoby rozpoznávání emocí

Na rozpoznávání emocí z obrazu se podrobněji zaměříme v této práci, nejprve si však uvedeme další možnosti rozpoznávání emocí.

6.1 Rozpoznávání emocí na základě fyziologických vlastností

Odkazovaná studie [14] měla za cíl zjistit, jaký z algoritmů pro strojové učení bude nejlépe určovat emoce z vlastností jako například EDA, elektrodermální aktivita, ECG – elektrokardiogram, teploty kůže a současného pulsu, získaného pomocí PPG - photoplethysmogramu, který pomocí optiky umožňuje okamžité získání pulsu. Na tyto údaje byly použity algoritmy SVM, LDF, CART, SOM a Naïve Bayes klasifikátor. Nejlepší výsledky měly SVM, a to 100% úspěšnost. SVM jsme v naší práci také použili, jsou popsány v podkapitole 9.3. Nejnižší úspěšnost měl LDA – pouze 50%. LDA jsme popsali v předchozí podkapitole. V jiných pracích byly testovány i další algoritmy, jako třeba K-NN. O K-NN si také povíme v podkapitole 9.3.

6.2 Rozpoznávání emocí podle hlasu

Při rozpoznávání emoci z hlasu se výzkumníci zaměřili na různé parametry řečového signálu, mezi které se řadí výška, síla (intensita, s jakou je promlouváno) a základní frekvence. Ke klasifikaci byly užity algoritmy SVM, SMO – Sequential Minimal Optimization, Decision Trees a K-NN. Ve studii [15] bylo zjištěno, že nejlepšími algoritmy pro tento druh úlohy jsou SVM a SMO.

6.3 Rozpoznávání emocí z textu

Emoce jsou z textu rozpoznávány na základě jeho syntaktické a sémantické struktury. V odkazované studii jsou získaná data ohodnocena pomocí databází jako WordNet a následně jsou klasifikována pomocí rozličných algoritmů, mezi které se řadí K-NN, PMI – Point Mutual Information a nebo jeho vylepšená verze, PMI-IR – PMI se získáváním informací. Tento přístup se prokázal jako účinnější než současné algoritmy pro strojové učení. Průměrná úspěšnost při určování dosáhla 84%. [16]

7 Použité datové sady emocí

Trénování neuronové sítě probíhalo z více datových sad, které si nyní blíže charakterizujeme.

7.1 Cohn-Kanade

Jako první byla použita databáze Cohn-Kanade (CK)[17], která obsahuje 486 sekvencí tváří 97 různých lidí. Každá ze sekvencí začíná neutrální emocí a končí snímkem, kdy je výraz odpovídající dané emoci nejsilnější. Použitelné pro trénování emocí tedy byly většinou jen poslední 3 snímky. Datová sada obsahuje 6 emocí – vztek, znechucení, strach, štěstí, smutek a překvapení. Další datové sady mají navíc ještě neutrální emoci.

Datová sada CK obsahuje černobíle obrázky. Ze všech obrázků jsem vyřezal pouze tváře pomocí OpenCV2 a kaskádového klasifikátoru. (způsob popsán v podkapitole 5.2). Získané tváře byly ořezány na 256x256 pixelů. V této fázi je v trénovací sadě 1635 souborů a v testovací sadě 407 souborů. Do testovací sady byly vybráno 20% náhodných obrázků z původních 2042 obrázků. Některé subjekty nemají kompletní nafocenou kompletní škálu emocí, jak lze vidět níže na příkladu z datové sady (6 je maximální počet emocí jednoho subjektu, neutrální emoce není obsažena)

Obrázek 5- datová sada CK, příklady emocí

První natrénovaná síť měla přesnost při testování 93% a při trénování byly použity konfigurační soubory z příkladů k datové sadě mnist dostupné ve frameworku Caffe. Úspěšnost testování je vcelku vysoká jenom z toho důvodu, že při testování i trénování byly použity obrázky pouze z datové sady CK – každý subjekt měl v naší datové sadě 3 snímky od stejné emoce, tudíž nebyla přílišná variace v tvářích.

7.2 AMFED

Další použitá databáze je AMFED, Affectiva-MIT Facial Expression Dataset [18], která obsahuje 242 videí zabírajících tváře účastníků. Celkově je databáze tvořena 168359 snímky. Videa nebyla natáčena v laboratorních podmínkách, ale většinou webkamerami notebooků jednotlivých účastníků.

Statistická data AMFED databáze potřebná pro trénování naší sítě jsou uložena ve formátu csv. Tyto statistiky obsahují údaje jako expresivnost tváře, sílu úsměvu a mnoho dalších, označené časovým údajem určujícím pozici, na které byla tvář nalezena. Data jsem pomocí Pythonu načetl a poté pomocí knihovny OpenCV zahájil práci se samotnými video soubory, ze kterých jsem ukládal individuální snímky. V csv tabulce byly pouze čistě časové údaje, tudíž jsem musel násobit snímkovou frekvencí, v tomto případě 25, abych získal přibližné číslo snímku. Po získání pozice ve videu jsem snímek uložil,

ale jen tehdy, pokud procentová hodnota „síly“ úsměvu přesahovala v tabulce aspoň 20%. Tím jsem získal surové snímky z videí lidí natáčejících se webkamerou – ve většině snímku bylo obsaženo zbytečné pozadí. Použil jsem další skript v jazyku python, kde jsem pomocí zmíněného kaskádového klasifikátoru Haar vlastností ořezal jednotlivé snímky pouze na oblast, ve které se vyskytuje tvář. Z takto získaných obrázků jsem dále vytvořil seznam obsahující jméno souboru a číselnou hodnotu značící emoci – v tomto případě 3 pro úsměv, štěstí. Tento soubor je použit ve frameworku Caffe. Obrázky menší než 100x100 byly vymazány z důvodu pozdější změny velikosti na 190x190, rozměru, který byl stanoven jako společný pro všechny vstupní obrázky. Obrázky s nižším rozlišením byly pro účely trénování nevhodné. Z původních 3176 fotek usmívajících se lidí zůstalo 2240 položek. 20% (448) náhodných fotek z tohoto množství bylo vybráno na testování a ze zbylých 1792 pak 5% (89) pro validaci.

S AMFED databází jsem spojil již připravené datové sady KDEF, CK a SFEW2. Celkem má trénovací sada 5013 obrázků a validační 1481 obrázků. Data byla při převedení do formátu lmdb převedena na velikost 192x192 a zamíchána.

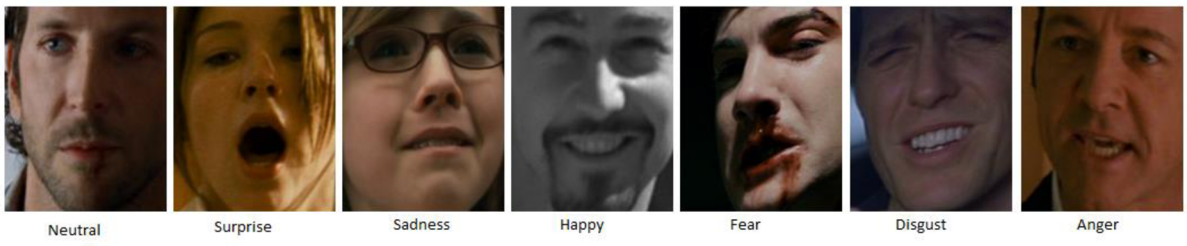
7.3 KDEF

Další zmíněná datová sada je KDEF, Karolinska Directed Emotional Faces [19], který obsahuje 4900 snímků projevů emocí ve tváři. Snímky jsou barevné, celkem jsou to fotografie 70 účastníků všech věkových kategorií. Každý z účastníků projevuje 7 různých emocí, které jsou nafoceny z 5 různých úhlů. Do naší datové sady extrahuji pouze snímky focené zepředu a na všechny takto získané snímky opět pouštím detektor tváří zepředu. Pro trénování získávám 785 snímků a pro testování 196 snímků. Snímky převádím na 256x256.

7.4 SFEW

V naší datové sadě byla použita data také z SFEW, Static Facial Expressions in the Wild [20], která obsahuje anotované snímky tváří s emocemi získaných z filmů. Snímky byly původně různě deformované, obdrženy z různých úhlů, výzkumníci ale tváře zarovnali. Naše datová sada pro trénovaní obsahuje 890 snímků a pro testování 431 snímků z datové sady SFEW.

V datové sadě SFEW2 se vyskytovaly i výrazy tváře snímané ze strany. Nevhodné snímky byly eliminovány pomocí skriptu pro detekci pohledu zepředu. Při trénování se díky zdeformovaným tvářím zastavila úspěšnost testování na 77%. Naše část datové sady převzaté z SFEW původně obsahovala 891 položek pro trénování, zbylo 461 po detekci tváří zepředu. Pro testování zůstalo ze 491 položek jen 209. Příklady jednotlivých emoci z datové sady SFEW na obrázku 6.



Obrázek 6 - datová sada SFEW, příklady všech emocí v datové sadě

7.5 Brazilian FEI

Další použitá databáze je Brazilian FEI [21], brazilská databáze tváří, která obsahuje 14 snímků každého z 200 účastníků, dohromady 2800 snímků. Všechny obrázky jsou barevné a účastníci na nich stojí před bílým pozadím. Prvních 10, někdy11 fotografií je pořízeno na celé škále úhlu 180. účastníci jsou studenti fakulty FEI a jejich věk je od 19 do 40 let. Počet žen a mužů je ve stejném poměru. Je obsažen neutrální a veselý výraz, většinou 12 až 15 fotografií. Skriptem byla extrahována pouze čtveřice 11, 12, 13 a 14, kde 11 je neutrální výraz, 12 je veselý, usměvavý výraz a 13 se 14 jsou neutrální výrazy v pozměněném osvětlení. Skriptem vyberu pouze tyto čtveřice a následně z nich vyjmu pouze ty se zepředu rozpoznatelnou tváří.

Proběhla také zběžná manuální kontrola obrazových dat a následné smazání fotografií subjektů, kteří nepochopili, že při snímání neutrálního výrazu se nemají usmívat.
Na obrázku lze vidět, že použitelné obrázky do naší datové sady jsou vždy jen poslední 4.

Obrázek 7- datová sada Brazilian FEI

Celkem po roztřídění a filtrování podle pohledu zepředu zůstalo 741 souborů, z toho cca 75% neutrální výraz a 25% štěstí. 20% (148) bylo náhodně zvoleno do testovací sady.

7.6 Vlastní datová sada

V této fázi vývoje je k dispozici 5175 obrázků k trénování a 1407 k testování. S těmito daty trénuji Caffe model, dosahuji přesnosti 85% při testování sítě. Při manuální kontrole klasifikace je však úspěšnost mnohem menší, většina obrázků je určena jako štěstí. Zmenšuji tedy část datové sady s obrázky štěstí o 70% a navíc je na všechny obrázky použit test tváře zepředu. Všechny obrázky jsou převedeny na černobíle pro menší velikost dat. Po profiltrování je natrénování k dispozici 2704 obrázků a k testování 725, což není mnoho. Uchyluji se proto k vytvoření vlastní datové sady, která bude obsahovat položky přímo z trailerů.

Další použité datové sady již nebyly stažené z internetu, ale vlastnoručně vytvořeny. Z imdb databáze stahuji 56 náhodných trailerů z žebříčku 100 nejpopulárnějších filmů. Na všech snímcích z filmových trailerů je provedena detekce tváře zepředu. Celkem bylo detekováno okolo 27000 tváří. Příliš malé obrázky – pod 64px - jsem odstranil. Nyní je v datové sadě okolo 16000 obrázků. Pro manuální anotaci jedním člověkem je to stále značné množství. Vytvářím proto skript pro detekci podobných obrázků. K tomu používám Python knihoven imagehash a difflib.

Způsob detekce podobnosti je následující – obrázky jsou v sekvenci za sebou ve většině případů. Proto jsou procházeny jednotlivě za sebou, pro každý obrázek je spočítán imagehash a následně je hash

tohoto obrázku porovnán s hashem obrázku následujícího. Pokud se hash shoduje na víc jak 60%, obrázek mažu. Tímto byla vlastní datová sada zmenšena na 9772 obrázků.

Při anotaci zjišťuji, že mnoho emocí neodpovídá ani jedné z dostupných k označení, ale spíš by spadaly do kategorie bolest nebo zmatení. V trailerech také není téměř žádná emoce strachu nebo znechucení. Pro anotaci také zběžně zkouším použít Microsoft Emotions API[5], klasifikace je ale velmi nepřesná.

Do datové sady přidávám emoce z dalších filmů, emoce znechucení je hodně obsažena například ve filmu Interview a strach přidávám z hororů Rec 1 a 2.

V ručně anotované datové sadě je přebytek neutrální emoce, čemuž se budeme dále věnovat v kapitole 8 - Experimenty.

[5] https://www.microsoft.com/cognitive-services/en-us/emotion-api

8 Experimenty

8.1 Ladění neuronové sítě

V ručně vytvořené datové sadě je přebytek neutrální a veselé emoce. Náhodným výběrem zmenšuji zmíněné subsety na 20% původní velikosti. Viz tabulku

Emoce	**Počet položek**
Anger	250
Disgust	101
Fear	77
Happy	545 → 100
Sad	269
Surprised	250
Neutral	1337 → 260

Tabulka 1- počty položek v datové sadě

Do datové sady přidávám zpět veškerá data ze SFEW, předtím naše datová sada obsahovala pouze tváře, které prošly filtrem detekce obličeje zepředu. Nyní obsahuje tedy i tváře, které jsou různě pokřivené nebo jsou zabrány z různých úhlů.

V tento okamžik obsahuje naše datová sada 2840 položek. Přidávám do ní 1314 položek z manuálně vytvořené datové sady. Po menších úpravách a přidání některých snímků anotovaných z dalších trailerů obsahuje trénovací datová sada 4139 položek a testovací 1164 položek.

Datovou sadu připravenou k experimentům pojmenovávám Alpha a vytvářím pomocí skriptů poskytnutých frameworkem Caffe soubory LMDB (popsáno v sekci Caffe). Trénuji s pomocí strojů Metacentra [22].

Statistiky pro nynější datovou sadu train-test.

emoce	TRAIN	TEST
Angry	620	145
Disgusted	432	110
Fear	301 → 337	108
Happy	767	257
Sad	520	124
Surprised	808	232
Neutral	689	186

Tabulka 2 - počty položek, Alpha

Subset emoce strachu má menší objem v porovnání s ostatními, manuálně proto anotuji a přidávám tváře získané z dalších trailerů horrorových filmů. Přidávám tváře z hororů It follows, Alien 3, Identity, The conjuring, The silence of the lambs, Dawn of the dead, Insidious, Predator, Sinister, The Thing (verze 1982 i 2011), The silence of the lambs. Z těchto 11 filmů je získáno pouze 36 použitelných tváří s emocí strachu. Subset strach proto obsahuje 337 položek.

Zahajuji trénování Caffe sítě, testy nevykazují přílišnou úspěšnost. Normalizuji proto datovou sadu odstraněním podobných obrázků (shoda nad 20%).

Nová testovací datová sada

Surprise	388
Neutral	371
Sadness	398
Happy	394
Disgusted	368
Angry	320
Afraid	335

Tabulka 3- dataset Alpha, v2, data pro Bravo a Charlie

Rozměry jsou při vytváření LMDB změněny na 64x64 pixelů. Nastavuji learning rate 0.001. Po pár tisících iterací je přesnost testování přibližně 60%. Měním learning rate na 0.01 a dostávám se na 63%.

Vytvářím více verzí Bravo a Charlie s mírně pozměněnými parametry a testuji jak automaticky, tak manuálně na malé testovací sadě tváří z Harry Potter trailerů. Úspěšnost je u všech víceméně stejná.

Vytvářím datovou sadu s kódovým označením Delta, do které jsem přidal i zrcadlené obrázky (parametr mirror: true v konfiguračním souboru Caffe z nějakého důvodu nefungoval, vytvářím a přidávám proto zrcadlené obrázky manuálně). Celkem je v trénovací sadě 5148 položek. Konfigurační soubor[6] pro model používám z bvlc_reference_caffenet, dostupný ve frameworku Caffe.

Dalšímu modelu dávám název Echo. Obrázky konvertovány na 128x128px. Data jsou stejná jako při fázi Delta. Přesnost při testování se dostává na 63% při iteraci 6580. Výsledky jsou dostačující, nyní manuálně porovnávám model Alpha a model Echo na obrázcích z trailerů Harry Potter.

Podle mého názoru jsem ohodnocoval správnost klasifikace emocí modely Alpha a Echo. Alpha klasifikovala správně v 34% případů a Echo v 66% případů. Jako finální model, v podstatě základní prvek naší aplikace, tedy volím model Echo.

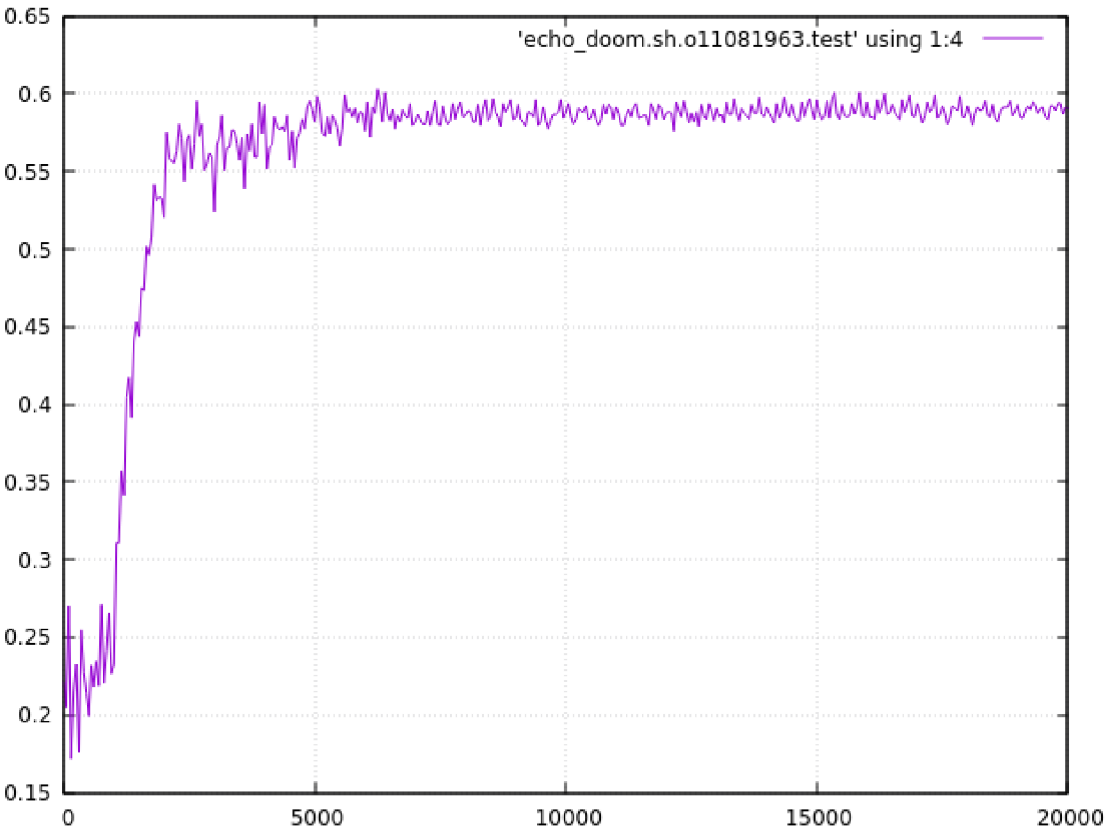
Níže je uvedena tabulka confusion matice pro síť echo, kterou budeme v naší aplikaci používat.

	anger	disgust	fear	happy	sadness	surprise	neutral	accuracy
anger	63	8	13	7	11	7	36	57,80%
disgust	3	72	5	14	5	0	11	72,73%
fear	7	1	45	16	10	9	20	51,14%
happy	7	14	14	191	6	6	20	80,25%
sadness	19	10	17	15	36	2	25	36,36%
surprise	7	10	28	5	15	144	23	68,90%
neutral	15	10	40	15	13	20	73	64,60%

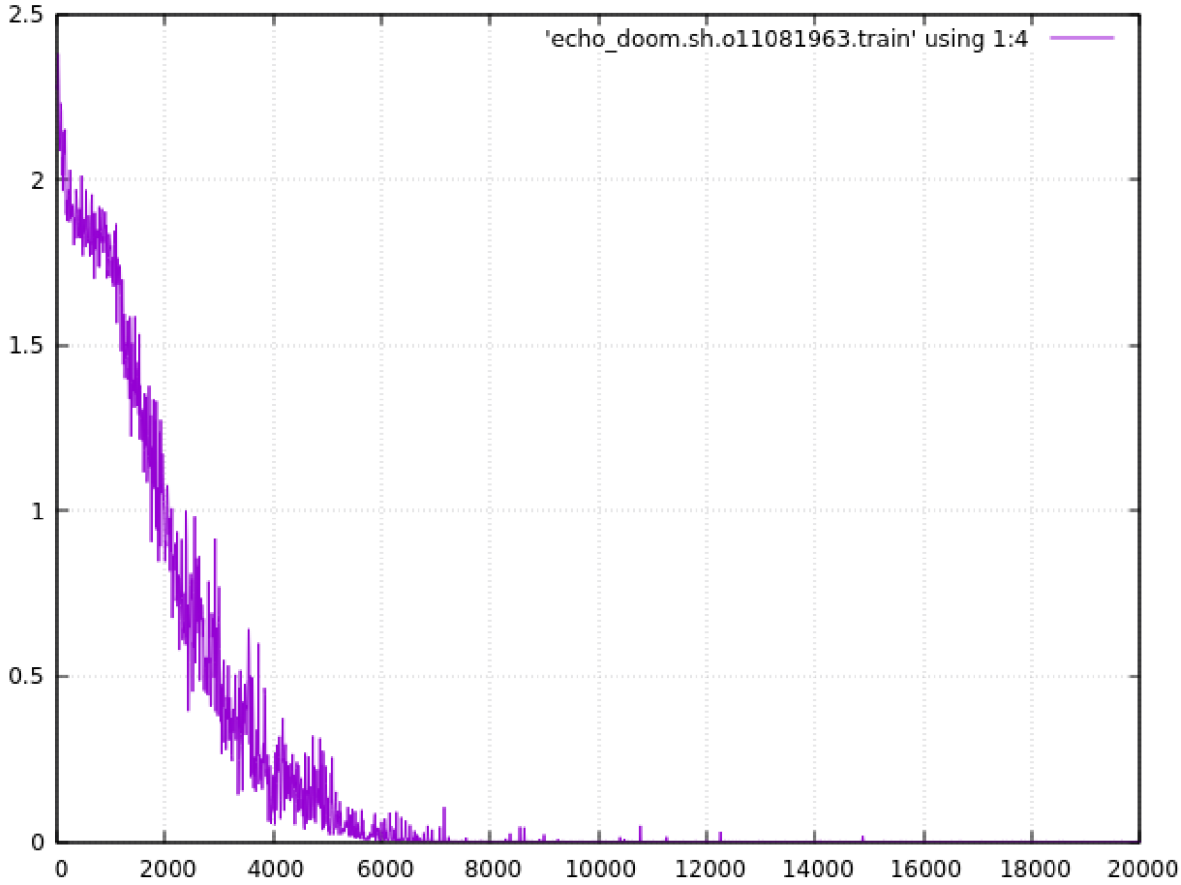
Tabulka 4- Confusion matice finální sítě

Výstupem při trénování pomocí Caffe jsou i statistiky o přesnosti při testování a výstupu loss funkce, jež můžeme vidět na grafech níže. Trénování proběhlo na vzdálených grafických strojích Metacentra.

[6] models/bvlc_reference_caffenet/train_val.prototxt



Obrázek 8- úspěšnost detekce při testování, y accuracy, x iterace



Obrázek 9- loss funkce při testování, y loss, x iterace

9 Tvorba datové sady pro rozpoznávání žánrů pomocí emocí

9.1 Existující práce o rozpoznávání žánrů

Existující výzkumy rozpoznávání emocí obsahují například rozpoznávání žánru pomocí zvukové stopy nebo kombinací audiovizuálních prvků. K tomuto jsou využity neuronové sítě ve všech níže uvedených výzkumech. Výzkumy jsou však pro kompletní filmy, tzn., že mají k dispozici nepoměrně více dat na učení a klasifikaci. Výzkum pro určování žánru podle zvuku navíc určuje pouze binárně – je film akční nebo není. [23]

Další výzkum používá rozmanité údaje – určuje žánr filmu podle audia, tak i videa. Jsou detekovány všechny možné informace, saturace, délka přechodů mezi scénami, kontrast a navíc jsou detekovány i objekty ve videu. Detekce je prováděna taktéž na celých filmech. Úspěšnost této metody je okolo 90%. [24]

Jiný existující výzkum naproti tomu určuje binárně – určuje, zda je film horor nebo akce. Úspěšnost je také vysoká, ve výzkumu se používá detekce objektů a charakteristik videa. [25]

Moje práce využívá pouze krátkých, většinou 2 minutových upoutávek na filmy, jelikož trénování na filmech by vyžadovalo získání legální kopie, mnohem větší výpočetní výkon a bylo by velmi časově náročné.

V následujících kapitolách popíšu proces, který vedl ke konečnému rozpoznání žánrů z traileru, nejprve se ale podíváme na to, co nám trénování vůbec umožnilo a poté na samotný způsob implementace a použité algoritmy.

9.2 Získání dat z trailerů

V následující kapitole popíši, jak jsem postupoval při učení modelu pro rozpoznávání žánru traileru z emocí v něm obsaženém.

Nejprve jsem stáhnul 100 trailerů od každého z hlavních žánrů. K tomuto jsem použil IMDB databázi a její žebříček nepopulárnějších filmů s alespoň 1000 hlasy, roztříděných podle žánru. Ke stahování používám open-source program youtube-dl [26], volně dostupný z repositářů většiny linuxových distribucí. Trailery stahuji v rozlišení 360p ve formátu mp4. Trailery hledám na Youtube, vyhledávám je podle názvu a roku jejich natáčení, stahuji první nalezené video. Z 600 videí je pouze 7 chybných, tato videa mažu.

Níže uvádím počet tváří nalezených v trailerech jednotlivých žánrů. K detekci tváří byl opět použit detektor používající kaskádové klasifikátory Haar vlastností. Každé z videí procházím po 8 snímcích, a pokud je nalezena tvář, ukládám ji. Pro každý takto získaný soubor ukládám číslo snímku videa, ve kterém byl nalezen a celkový počet snímků videa. Tato data později využiji pro výpočet různých statistik.

Žánr	Počet nalezených tváří
Horor	7716
Komedie	14785
Drama	8947
Thriller	7690
Romance	13075
Akce	5867

Tabulka 5- počet nalezených tváří podle žánrů

Nyní začínám s klasifikací emocí pro jednotlivé žánry, používám k tomu c++ program classification.bin dostupný ve frameworku Caffe a spouštím na klasifikaci na vzdálených strojích Metacentra.

Následuje tabulka rozložení roku natáčení filmů, resp. jejich trailerů v původní trénovací sadě.

Pořadí	Rok	Počet	Procenta
1.	2015	197	32.8333
2.	2016	80	13.3333
3.	2014	68	11.3333
4.	2013	33	5.5000
5.	2012	20	3.3333
6.	2010	15	2.5000
7.	2004	14	2.3333
8.	2009	13	2.1667
9.	2006	12	2.0000
10.	2005	12	2.0000
11.	1994	12	2.0000
12.	2008	10	1.6667
13.	2007	10	1.6667
14.	2011	10	1.6667
15.	1999	9	1.5000
16.	2002	7	1.1667
17.	2001	6	1.0000
18.	1985	6	1.0000
19.	1991	6	1.0000
20.	2003	5	0.8333
21.	1987	5	0.8333
22.	1996	5	0.8333
23.	2000	4	0.6667
24.	1993	4	0.6667
25.	1982	3	0.5000
26.	1989	3	0.5000
27.	1995	3	0.5000
28.	1997	3	0.5000
29.	1998	3	0.5000
30.	1992	3	0.5000
31.	1979	2	0.3333
32.	1980	2	0.3333
33.	1986	2	0.3333
34.	1984	2	0.3333
35.	1988	2	0.3333
36.	1961	2	0.3333
37.	1972	1	0.1667
38.	1973	1	0.1667
39.	1977	1	0.1667
40.	1978	1	0.1667
41.	1983	1	0.1667
42.	1967	1	0.1667
43.	1960	1	0.1667

Tabulka 6 - rozložení trénovací sady podle roku natáčení traileru

9.3 Použité algoritmy

Jako první algoritmus jsem vytvořil svůj vlastní. K tomu jsem potřeboval nejprve vytvořit statistiky. V následujícím textu je popsán proces jejich získání.

Pro každý obrázek získávám celkový počet snímků videa, ve kterém byl nalezen a toto číslo převádím na časový údaj v sekundách – dělením 30, zaokrouhluji nahoru. Vytvářím pole délky stejné jako počet právě získaných sekund. Pro každý obrázek dále zjišťuji číslo snímku, ve kterém byl detekován a převádím také na časový údaj v sekundách, zaokrouhluji dolů a ukládám na příslušný index pole. Z celkové délky pole vypočítávám procenta náležící jednomu poli. Tváře uložené v jednom poli se dále rovnoměrně dělí o tato procenta.

Dále provádím výpočet celkové délky, jakou byly v traileru zobrazeny emoce a ukládám jak tuto délku, tak poměr jednotlivých emocí v tomto čase.

Pro testování stahuji 500 trailerů, rozdílných od těch, na kterých probíhalo trénování. Testování shodnosti je uskutečněno podle IMDB ID filmu daného traileru (imdb id v názvu každého z video

souborů, včetně extrahovaných obrázků). Na počátku byl k videu ukládán pouze jeden žánr, podle toho, z jakého žebříčku populárnosti bylo video staženo. Při testování je však pouze jeden žánr nedostačující. Vytvářím Python skript, který používá imdbpy knihovnu pro komunikaci s imdb api a pro každý z trailerů stahuji všechny žánry, do kterých daný film patří. Některé filmy mají například i 5 žánrů.

9.3.1 Vlastní algoritmus

Můj algoritmus používá bodový systém. Pro každou z emocí existuje tabulka, jež obsahuje parametry jednotlivých žánrů, tedy poměr počtu nalezených tváří vůči délce videa a procentové zastoupení emocí v tomto čase. Tyto údaje byly vypočítány jako průměr ze všech trénovacích trailerů, proto je databáze tvořena 6x8 hodnotami (6 žánrů, 1 tabulka pro screentime, 7 pro emoce).

Při detekci zjišťuji totožné údaje o testovaném videu a následně porovnávám každý z parametrů s existujícími daty. První nejbližší údaj je ohodnocen jedním bodem, druhý nejbližší půlbodem.
Příklad:
Detekované video v sobě obsahuje tváře 40% času, z toho jsou 90% šťastné a 10% neutrální. Podle tabulky 8 s použitými statistikami je těmto údajům nejblíž Komedie s 40.86% screentime (+1 bod), na druhé pozici Romance s 39.42% screentime (+0.5), podle emoce štěstí je na prvním místě Romance (+1) a na druhém Komedie (+0.5). Podle neutrální emoce je na prvním místě Komedie (+1) a na druhém Drama (+0.5). Tím pádem výsledek vypadá takto: Komedie 2.5 bodu, Romance 1.5 bodu a Drama 0.5 bodu.
V další verzi jsem experimentoval s dalším parametrem, a to průměrnou sílou jednotlivých emocí. Použil jsem dalších 7 tabulek, testy ale ukázaly, že úspěšnost se zhoršila o několik procent. Následuje tabulka s porovnáním:

Verze testu*	Verze se sílou	Verze bez síly
top1	36%	39%
top3	75%	86%
top5	95%	96%

*Tabulka 7- porovnání úspěšností algoritmů, s parametrem síly emoce a bez; *top1 – výsledek na prvním místě, top3 – mezi prvními 3, top5 - mezi prvními 5; **Výsledky porovnávány se všemi možnými žánry daného filmu podle Imdb*

Průměrné statistiky použité pro první algoritmus (čísla značí procenta, značící poměr počtu detekovaných tváří vůči délce videa (tabulka SCREENTIMES) a jednotlivé emoce obsahují procenta z celkového screentime). Průměr je z 600 trénovacích trailerů, 100 pro každý žánr. Tabulky se statistikami jsou uvedeny na následující straně.

SCREENTIMES	
COMEDY	40.86478301224219
ROMANCE	39.42529846606875
DRAMAS	31.33337726395317
THRILLER	27.28839131252104
HORRORS	27.232985979084
ACTION	21.471169261278178

SADNESS	
COMEDY	14.614463622153457
ROMANCE	13.812773997625879
DRAMAS	12.4595175758768
ACTION	11.571526064996611
THRILLER	10.678630785290709
HORRORS	10.399620832338218

NEUTRAL	
THRILLER	38.8631379374622
HORRORS	36.756088593349055
ROMANCE	35.568895286744215
ACTION	35.329854103722354
DRAMAS	34.454183024349945
COMEDY	31.0284778602759

ANGER	
ACTION	13.10306587548661
DRAMAS	12.368134869684434
THRILLER	11.123927493393895
COMEDY	10.950168311943754
HORRORS	10.2056856604614
ROMANCE	8.56689594933217

FEAR	
HORRORS	20.156421883058197
THRILLER	16.00652763848753
ACTION	15.263836715138249
DRAMAS	15.185356416861703
ROMANCE	13.06630639952647
COMEDY	12.146523643742846

DISGUST	
COMEDY	6.0039856412870725
THRILLER	5.972129704901479
HORRORS	5.74527537522032
DRAMAS	5.6133651805461895
ACTION	5.574035439036239
ROMANCE	4.529041522363037

HAPPY	
ROMANCE	18.826768762634323
COMEDY	18.14009518263926
DRAMAS	15.018731097072699
ACTION	13.522426958577395
HORRORS	12.774362553690935
THRILLER	12.723234330446322

SURPRISED	
COMEDY	7.116285737957701
ACTION	5.6352548430425315
ROMANCE	5.629318081773923
DRAMAS	4.900711835608245
THRILLER	4.632412110017885
HORRORS	3.962545101881857

Tabulka 8-použitá statistická data pro algoritmus, data jsou výsledkem trénování

9.3.2 Support Vector Machines – Algoritmy podpůrných vektorů

Jako další algoritmus zkouším Support Vector Machines. Využívám programu libSVM [27] [28].

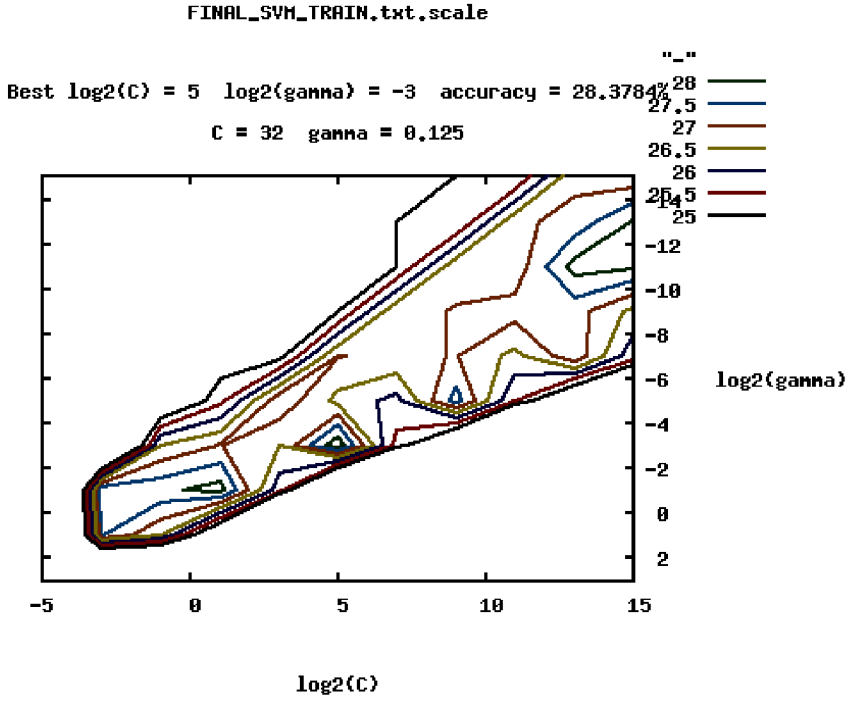
SVM hledají optimální nadrovinu, která rozděluje více tříd tak, aby její vzdálenost od bodů určujících okraj jednotlivých tříd byla co největší. Body ležící na okraji jsou nazývány podpůrné vektory [29].

Při trénování svm jsem použil stejná data jako ve vlastním algoritmu, tzn. screentime a poměr jednotlivých emocí ku celkovému screentime. V této fázi již nepoužívám 600 trailerů, ale jen 593, byly totiž nalezeny vícežánrové duplikáty. S tímto algoritmem není použit pouze průměr, ale všechny statistiky jsou vypočítány pro všech 593 trailerů a následně je na nich trénován svm model.

K trénování jsem použil program svm-train, který je volně dostupný ke stažení. Je vytvořen soubor model, který následně používám pro testování programem svm-predict. Existuje ještě program svm-scale, který automaticky upravuje parametry svm modelu. Používám ho ve druhé verzi modelu, první verzi nyní popíšu.

V první verzi porovnávám pouze první výsledek testu, netestuji všechny potenciální žánry pro daný trailer, pouze první určený s žánrem filmu odpovídající žebříčku na imdb, ve kterém byl trailer pro původní testovací sadu nelezen. Tento způsob má přesnost pouze 24.8497%.

V druhé verzi již testuji všechny možné žánry traileru a dostávám se na úspěšnost 33%. Na tuto verzi zkouším použít již zmíněny program svm-scale, který upraví parametry učení. Graf znázorňující změny parametrů při trénování uveden níže na obrázku 10.



Obrázek 10- trénování škálované SVM

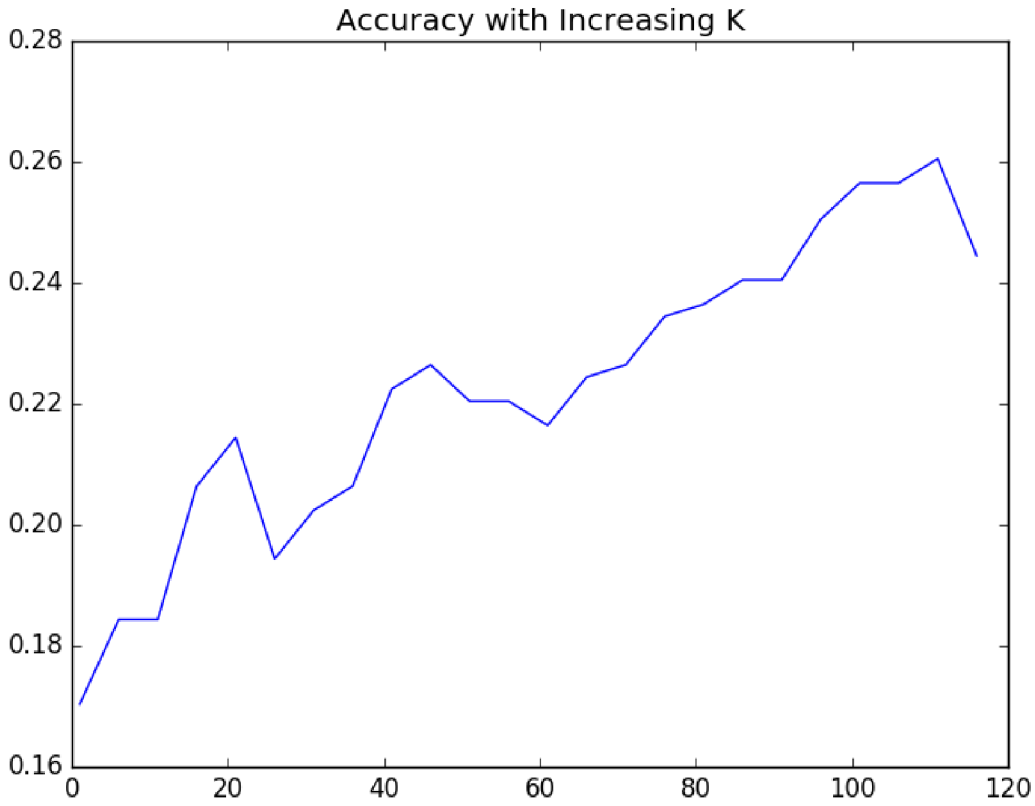
Po použití svm-scale se přesnost klasifikace zvyšuje na 37%.

9.3.3 K-nearest neighbors – algoritmus k-nejbližších sousedů

Jeden ze základních algoritmů pro klasifikaci. Obecně je při klasifikaci vypočítána euklidovská vzdálenost mezi testovaným bodem a trénovacími daty, resp. body ve vzdálenosti menší než K od klasifikovaného bodu. Třída, která má v okolí nejvíce bodů je výsledkem klasifikace [30].

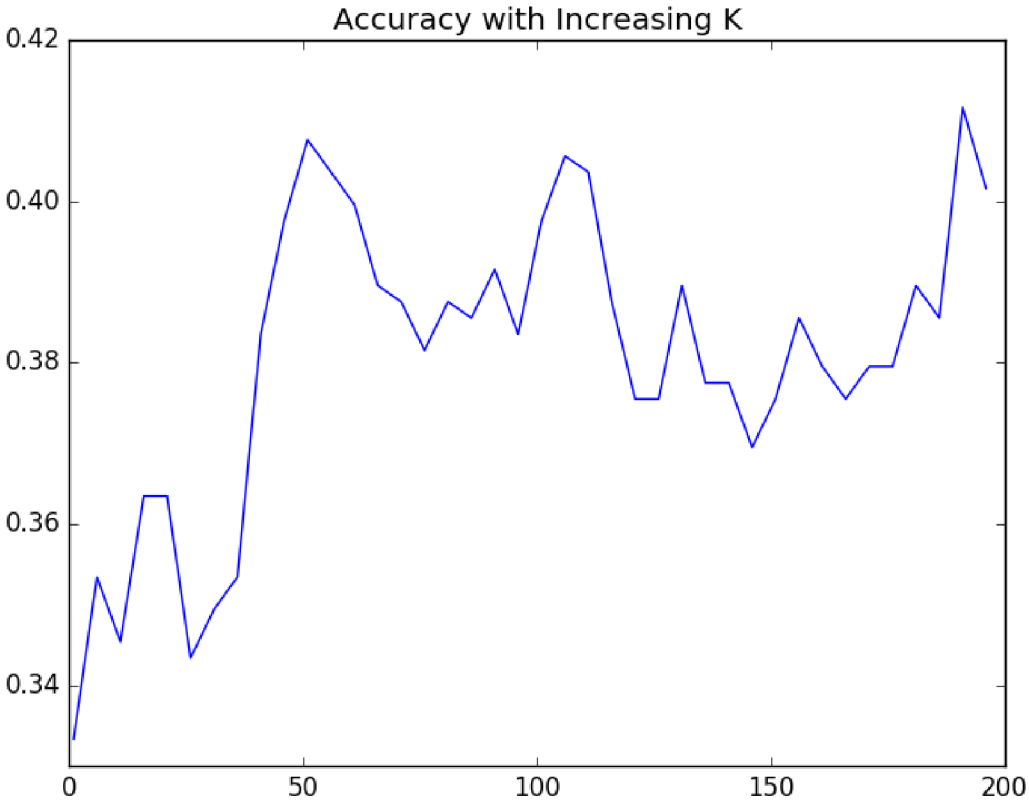
V mém programu jsem použil knihovnu scikit.learn [31] pro python. Standardní používaná metrika algoritmu K-NN je zde Minkowskiho, parametr p=2, což je identické s Euklidovou.

Při mém trénování jsem při prvním pokusu testoval úspěšnost pouze pro první nalezený žánr, přesnost dosahovala maximální hodnoty pro K=111, a to 26%, jak je také možno vidět v grafu na obrázku 11.



Obrázek 11- přesnost určování se vzrůstajícím parametrem K

V druhé verzi jsem již testoval všechny možné žánry testovaného traileru, pro K=111 dosahuji přesnosti 40%. Použil jsem stejné K jako při testování pouze prvního nejlepšího, nyní testuji všechny dostupné žánry, hledám proto nejlepší K znovu. Zjišťuji, že v tomto případě je nejlepší K=53 a K=191. Používám K=53. Takto dosahuji průměrné přesnosti 41.7%, což je nejvíc ze všech použitých klasifikačních algoritmů.



Obrázek 12- K-NN, parametr K, testování s více žánry naráz

Testovací data rozděluji podle let, jelikož trénovací sada obsahovala většinu trailerů z roku 2014 a novější, kdežto testovací sada obsahuje většinu trailerů z roku 1960 a starších, tím pádem je přesnost hodně ovlivněna rokem natáčení testovaného traileru. Datovou sadu rozděluji na filmy před rokem 1980, na filmy mezi lety 1980 až 2005 a na novější než 2005.
Při K=53 dosahuji následujících úspěšností

Rozmezí let	Úspěšnost
< 1980	39.18%
1980 – 2005	41.31%
> 2005	47.32%

Tabulka 9- výsledná úspěšnost při testování algoritmu podle roku natáčení filmového traileru

Je zřejmé, že čím novější film, tím lepší úspěšnost detekce žánru, což přímo souvisí s trénovací sadou, která obsahuje většinu trailerů na filmy z roku 2015, jak lze vidět v tabulce 6 v podkapitole 9.2. Starší trailery měly zcela jinou strukturu než ty ze současné doby, zpravidla se jednalo o pár klipů přímo vystřižených z filmu bez předělujícího textu, nebo celý trailer byl jediný klip z daného filmu. V dnešní době jsou trailery složeny z mnoha klipů kratšího charakteru, často jsou mezi nimi přechody, někdy zcela černé, někdy obsahující text s citacemi od filmových kritiků, někdy klíčová slova definující onen film, ale klíčové je, že obsahují zcela jiné poměry viditelných tváří vůči době trvání videa, než trailery staré.

9.4 Výsledky zvoleného algoritmu

Pro klasifikaci žánrů použijeme algoritmus nejbližších sousedů, poněvadž se prokázal jako nejlepší. Na další straně je matice záměn (confusion matrix) pro jednotlivé kombinace žánrů. Testováno bylo na položkách s více možnými třídami a výsledná tabulka pokrývá všechny kombinace.

Žánry traileru	horror	thriller	comedy	romance	drama	action	accuracy
thriller,drama	15	3	18	6	5	8	14,55%
romance,drama	8	3	15	8	5	9	27,08%
drama	8	6	10	5	6	5	15,00%
comedy,romance,drama	2	5	14	6	4	5	66,67%
horror	8	3	14	2	3	1	25,81%
comedy,drama	7	1	13	3	2	2	53,57%
thriller,action	6	4	6	6	4	1	18,52%
comedy	6	1	7	4	5	1	29,17%
comedy,romance	4	0	10	3	1	3	61,90%
thriller,drama,action	4	2	8	0	5	1	40,00%
horror,thriller	5	1	7	4	0	2	31,58%
action	5	2	4	5	2	1	5,26%
drama,action	3	1	5	4	1	2	18,75%
thriller	2	0	7	2	2	2	0,00%
horror,thriller,drama	5	1	4	1	1	1	53,85%
horror,drama	3	2	3	1	2	1	41,67%
horror,comedy	3	2	2	2	3	0	41,67%
thriller,romance,drama,action	3	0	2	2	0	1	37,50%
comedy,action	4	0	2	0	0	1	42,86%
horror,action	1	2	3	0	0	0	16,67%
romance,drama,action	2	1	1	0	1	1	33,33%
comedy,drama,action	2	1	3	0	0	0	50,00%
horror,thriller,action	2	0	2	1	0	0	40,00%
thriller,comedy,drama,action	0	0	4	0	0	0	100,00%
thriller,romance,drama	0	0	0	1	1	1	66,67%
horror,thriller,comedy	1	1	0	0	0	0	100,00%
thriller,comedy	0	0	2	0	0	0	100,00%
thriller,comedy,romance	1	0	0	1	0	0	50,00%
thriller,romance	0	0	0	0	1	1	0,00%
thriller,comedy,drama	0	0	1	0	0	0	100,00%
horror,comedy,romance	0	0	0	1	0	0	100,00%
horror,thriller,comedy,action	0	0	0	1	0	0	0,00%
horror,thriller,drama,action	0	0	1	0	0	0	0,00%
horror,comedy,drama	0	0	0	1	0	0	0,00%
horror,romance	0	0	1	0	0	0	0,00%
horror,romance,drama	0	0	1	0	0	0	0,00%
romance	0	0	0	0	0	1	0,00%
horror,thriller,comedy,drama,action	0	0	1	0	0	0	100,00%

Tabulka 10 - confusion matrix pro finální algoritmus, výsledky testování na 600 testovacích trailerech, multi-žánr

10 Popis aplikace a ukázka

Aplikace je psána v jazyce Python pro verzi 2.7. Aplikace nejprve načte vstupní video soubor, který pomocí OpenCV prochází po snímcích. Přitom probíhá detekce obličejů ve snímku pomocí kaskádového klasifikátoru za použití modelu uloženého v souboru *haarcascade_frontalface_alt.xml.*

Detekované tváře jsou uloženy ve složce s programem. Jako další přichází na scénu klasifikátor frameworku Caffe. Je volán systémovým voláním, jelikož přes interface Pythonu vykazovala klasifikace rozdílné výsledky, než přes binární soubor. Výsledky klasifikace jednotlivých snímků jsou uloženy a po dokončení zpracovány programem. Z výsledků jsou vytvořeny základní statistiky, jako doba zobrazení jednotlivých emocí/tváří vzhledem k délce videa atp., již bylo popsáno v předchozích kapitolách. Následně je provedena klasifikace pomocí algoritmu nejbližších sousedů a zjištěný žánr je spolu se statistikami vypsán na standardní výstup. Na obrázcích níže je demonstrována činnost programu.

```
[john@pc BP]$ python2.7 GenreDetection.py Mr.\ Robot\ Season\ 2\ Trailer\ \(HD\)-Oc-AsN7d1wg.mp4
Mr. Robot Season 2 Trailer (HD)-Oc-AsN7d1wg.mp4
Begin face detection...
18.3771767321
```

Obrázek 13 - Počátek detekce tváří, zobrazena procenta dokončení procházení videa

```
[john@pc BP]$ python2.7 GenreDetection.py Mr.\ Robot\ Season\ 2\ Trailer\ \(HD\)-Oc-AsN7d1wg.mp4
Mr. Robot Season 2 Trailer (HD)-Oc-AsN7d1wg.mp4
Begin face detection...
Classifying image 57 out of 68 (83.8235294118%)
```

Obrázek 14 - klasifikace obrázků s detekovanými tvářemi

```
[john@pc BP]$ python2.7 GenreDetection.py Mr.\ Robot\ Season\ 2\ Trailer\ \(HD\)-Oc-AsN7d1wg.mp4
Mr. Robot Season 2 Trailer (HD)-Oc-AsN7d1wg.mp4
Begin face detection...
genreId,anger,disgust,fear,happy,sadness,surprise,neutral,framescreentime
-1,0.0877192982456,0.105263157895,0.0877192982456,0.0701754385965,0.0175438596491,0,0.631578947368,0.0211189329381
RESULT_EMOTION_INDEX:4
Genre is drama
[john@pc BP]$
```

Obrázek 15 - Klasifikace žánrů dokončena, výpis statistik a výsledku

Výsledkem klasifikace je drama, níže obrázek 16 z IMDB.com dokazující správnou detekci.

Obrázek 16 – Imdb.com, Mezi žánry traileru patří drama, klasifikace úspěšná

11 Závěr

Emoce v této práci rozpoznávám pomocí konvolučních neuronových sítí. Při klasifikaci žánru traileru nejprve načtu trailer pomocí python knihovny OpenCV a snímek po snímku procházím video, každých 8 snímků je provedena detekce obličeje ve snímku pomocí kaskádového klasifikátoru Haar vlastností. Tato operace je velmi výpočetně náročná, zabere přibližně stejný čas detekovat obličeje jako je klasifikovat na emoce.

Po provedení detekce všech obličejů ve videu je použit framework Caffe, který je open-source knihovnou pro práci s konvolučními neuronovými sítěmi. Klasifikace je prováděna pomocí natrénovaného modelu konvoluční sítě, která je naučena z tisíců obrázků emocí.

Po klasifikaci je vytvořena statistika jednotlivých emocí ve videu vzhledem k celkovému trvání traileru, resp. jejich poměru k celkovému času, po který byly v traileru zobrazeny tváře. Klasifikace žánru pak probíhá pomocí algoritmu K-nearest neighbors s parametrem K=53. Tato hodnota byla zjištěna testováním jako nejvhodnější a nejpřesnější.

Při určování žánrů je tedy načtena tabulka s již zjištěnými hodnotami pro jednotlivé trailery. Počet trénovacích trailerů je 593, přičemž je v naší datové sadě 6 žánrů, tedy každý žánr obsahuje okolo 100 trailerů. Algoritmus K-NN pak najde nejvíc podobných trailerů podle zjištěných hodnot v detekovaném traileru a žánr, který byl v modelu tímto způsobem nalezen, je výstupem programu.

Experimenty jsme zjistili, že podle emocí lze zjistit žánr filmového traileru. Úspěšnost klasifikace je nejlepší pro filmy po roce 2005, jelikož většina trénování žánrů proběhla na filmech z roku 2015 (více než třetina filmů, na kterých se trénovalo, je z roku 2015). Úspěšnost klasifikace dosahuje 47%. Toto platí pro trailery, moji práci lze využít i pro vylepšení stávajících způsobů rozpoznávání žánrů z trailerů i filmů. Pokud by byla moje práce použita na více filmech plné délky, úspěšnost detekce by měla dosahovat větší úspěšnosti díky více dostupným datům. Větší úspěšnosti klasifikace by také mohla pomoci důkladnější detekce obličejů v traileru, detekování obličejů natočených do všech úhlů a jejich transformace na pohled zepředu. Aplikace by mohla být rozšířena o využití větší škály emocí a tím by detekce žánrů mohla být také vylepšena.

Literatura

[1] *convolutional networks @ cs231n.github.io* [online]. Dostupné z: http://cs231n.github.io/convolutional-networks/

[2] *Convolutional_neural_network @ en.wikipedia.org* [online]. Dostupné z: https://en.wikipedia.org/wiki/Convolutional_neural_network#/media/File:Max_pooling.png

[3] JIA, Yangqing, Evan SHELHAMER, Jeff DONAHUE, Sergey KARAYEV, Jonathan LONG, Ross GIRSHICK, Sergio GUADARRAMA a Trevor DARRELL. Caffe: Convolutional Architecture for Fast Feature Embedding. *Proceedings of the ACM International Conference on Multimedia* [online]. 2014, s. 675–678. Dostupné z: doi:10.1145/2647868.2654889

[4] *Symas Lightning Memory-mapped Database @ symas.com* [online]. Dostupné z: https://symas.com/products/lightning-memory-mapped-database/

[5] *Dropoutlayer Lasagne @ github.com* [online]. Dostupné z: https://github.com/Lasagne/Lasagne/blob/master/lasagne/layers/noise.py

[6] *index @ keras.io* [online]. Dostupné z: http://keras.io/

[7] *opendeep @ github.com* [online]. Dostupné z: https://github.com/vitruvianscience/opendeep

[8] *index @ torch.ch* [online]. Dostupné z: http://torch.ch/

[9] CAMPS, Octavia a S NARASIMHAN. PCA-based Object Recognition [online]. nedatováno. Dostupné z: http://www.cse.psu.edu/~rtc12/CSE486/lecture32.pdf

[10] *File:Fisherfaces @ www.scholarpedia.org* [online]. Dostupné z: http://www.scholarpedia.org/article/File:Fisherfaces.jpg

[11] *Eigenfaces @ upload.wikimedia.org* [online]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/6/67/Eigenfaces.png

[12] *Tutorial_Py_Face_Detection @ Docs.Opencv.Org* [online]. Dostupné z: http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0

[13] *haarfeatures @ docs.opencv.org* [online]. Dostupné z: http://docs.opencv.org/2.4/_images/haarfeatures.png

[14] JANG, Eun-hye, Byoung-jun PARK, Sang-hyeob KIM a Jin-hun SOHN. Emotion classification by machine learning algorithm using physiological signals. *... Proceedings of Computer Science & Information ...* [online]. 2012, roč. 25, s. 1–5. Dostupné z: http://www.icmlc.org/icmlc2012/001_icmlc2012.pdf

[15] CASALE, S., a. RUSSO, G. SCEBBA a S. SERRANO. Speech Emotion Classification Using Machine Learning Algorithms. *2008 IEEE International Conference on Semantic Computing* [online]. 2008, roč. 118, č. 13, s. 167–174. Dostupné z: doi:10.1109/ICSC.2008.43

[16] *articleDetails @ ieeexplore.ieee.org* [online]. 2013. Dostupné z: doi:10.1109/SSP.2012.6319793

[17] KANADE, Takeo a Jeffrey F COHN. Comprehensive database for facial expression analysis. *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on* [online]. 2000, s. 46–53. Dostupné z: doi:dx.doi.org/10.1109/AFGR.2000.840611

[18] DANIEL MCDUFF, RANA EL KALIOUBY, THIBAUD SENECHAL, MAY AMR, JEFFREY COHN, Rosalind Picard. Affectiva MIT Facial Expression Dataset (AM-FED):

Naturalistic and Spontaneous Facial Expressions Collected "In the Wild", IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2013.

[19] FLYKT, Anders. KDEF and AKDEF. *Psychology* [online]. 1998, s. 3–5. Dostupné z: http://www.emotionlab.se/sites/default/files/About KDEF.pdf

[20] DHALL, Abhinav, Roland GOECKE, Simon LUCEY a Tom GEDEON. Static Facial Expression Analysis in Tough Conditions : Data , Evaluation Protocol and Benchmark Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. *Database* [online]. 2011, s. 2106–2112. Dostupné z: doi:10.1109/ICCVW.2011.6130508

[21] *facedatabase @ fei.edu.br* [online]. Dostupné z: http://fei.edu.br/~cet/facedatabase.html

[22] *index @ metavo.metacentrum.cz* [online]. Dostupné z: https://metavo.metacentrum.cz/

[23] JAIN, Sanjay a R S JADON. Audio Based Movies Characterization Using Neural Network [online]. 2008, roč. 1, č. 2, s. 87–90. Dostupné z: http://www.researchpublications.org/ijcsa/issue2/2008-ijcsa-01-02-03.pdf

[24] JAIN, Sanjay K a R S JADON. Movies genres classifier using neural network. *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on* [online]. 2009, s. 575–580. Dostupné z: https://pdfs.semanticscholar.org/4e2f/9a5879720137eac7b89bd8032ee4e76d054b.pdf

[25] NAIK, Vilas. International Journal of Research in Advent Technology ACTION AND HORROR GENRE IDENTIFICATION OF MOVIES USING ARTIFICIAL NEURAL NETWORK International Journal of Research in Advent Technology [online]. 2013, roč. 1, č. 5, s. 359–365. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.403.7451&rep=rep1&type=pdf

[26] *youtube-dl @ github.com* [online]. Dostupné z: https://github.com/rg3/youtube-dl/

[27] LIN, Chih-Chung Chang and Chih-Jen. *libSVM A Library for Support Vector Machines* [online]. Dostupné z: https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[28] WILLIAMS, Chris. Support Vector Machines [online]. 2008, roč. 1, č. October, s. 1–8. ISSN 1613-9011. Dostupné z: doi:10.1007/978-0-387-77242-4

[29] ŽIŽKA, Jan (Mu V Brně). Support vector machines (SVM). *FI:PA034 Strojové učení (podzim 2006)* [online]. 2006. Dostupné z: http://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf

[30] *K-nearest_neighbor @ scholarpedia.org* [online]. Dostupné z: http://scholarpedia.org/article/K-nearest_neighbor

[31] *sklearn @ scikit-learn.org* [online]. Dostupné z: http://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Příloha A

Obsah CD

- Tato práce ve formátu pdf
- Natrénovaný model pro rozpoznávání emocí z obrázků ve formátu caffemodel
- Konfigurační soubor pro kaskádový klasifikátor tváří zepředu ve formátu xml
- Konfigurační soubory pro správnou funkci klasifikátoru Caffe
- Data pro algoritmus K-NN pro rozpoznávání žánrů ve formátu txt (csv)
- Hlavní program v jazyce Python, seznam souborů
- Video prezentace

Příloha B

Dokumentace k aplikaci, spuštění

- Nainstalovat všechny potřebné knihovny pro správný běh frameworku Caffe a framework Caffe, zkompilovat tak, aby byl dostupný binární soubor classification.bin. Změnit cestu v hlavním programu GenreDetection.py v proměnné caffeApp na cestu k tomuto souboru
- Nainstalovat Python 2.7 a knihovny pandas, scikit.learn, StringIO a numpy
- Připravit si nějaký mp4 video trailer, přednostně v rozlišení 360p pro rychlejší detekci
- Spustit aplikaci – první parametr je název video souboru