SGBT: Toward Coarse to Fine Cast Recognition on Movie Stills with Deep Learning

Shannon Lee
Group 27 ShiaGiBaLuanTrain
National Taiwan University
R07921001

Yu-Cheng Wen
Group 27 ShiaGiBaLuanTrain
National Taiwan University
R07921017

Yu-Ting Hsu
Group 27 ShiaGiBaLuanTrain
National Taiwan University
R06921012

I. Introduction

Objective: To search for a person in a large database with a single image. We are given an image of a target cast and some candidates(frames of a movie with person bounding boxes).

Dataset: The used data comes from 250 movies. (IMDB) For each movie, the main cast are collected as queries. The dataset is divided into train (125 movies), validation (25 movies), and test (100 movies) data. Only train and validation data are released with GT label, while the test data is accessible only to TAs (on kaggle).

Cast

Candidates

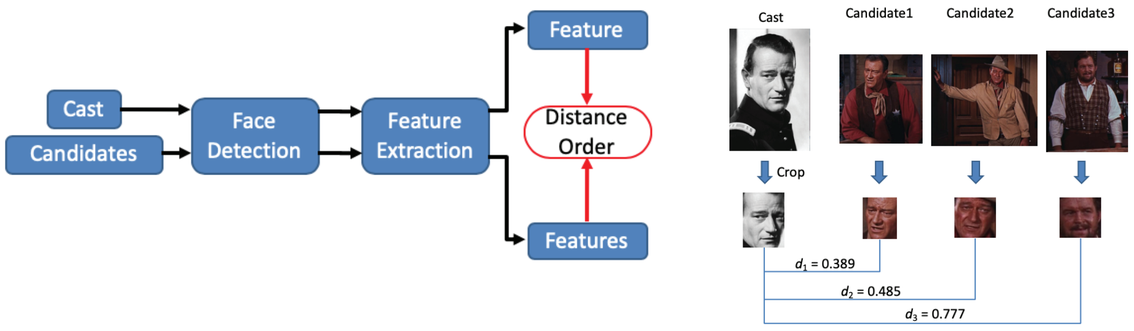
According to the tough condition of the given person bounding boxes(images), there may be someone's shoes, hand, back view, side face or anything else that could be recognized to the specific cast, not only limited in the face. Moreover, there even exists a category "others" makes the challenge more difficult.

So that we would not only focus on face recognition method but also depend on person re-id technique to complete our prediction. Using face recognition to make a coarse prediction, getting the distance between cast based on those candidates which faces could be found. Then we use the state-of-art person re-id network to encode candidates with well clustered features, and use the previous result to find the cluster belong to which cast. At the end of all, we could get the fine grinding prediction.

II. Face Recognition

The face is the most accurate element in the task of person identification since a person may change his/her wearings. Therefore, we rank the candidates with the feature of faces to boost the quality of cast feature extracted by the second-staged unsupervised feature extractor.

- In the preprocessing stage, we reorder the candidates by pre-trained feature extractor provided by Dlib library, a famous deep learning toolkit.
- The order is sorted by the L2-norm feature distances between candidates and the queried cast.
- Examples of the preprocess feature distances are shown below.



The architecture of the model that encodes our cropped faces are shown below.

layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

Network configuration. Details of the face CNN configuration A. The FC layers are listed as "convolution" as they are a special case of convolution in the network. For each convolution layer, the filter size, number of filters, stride and padding are indicated.

This model is pre-trained with about 3 million faces and reach a high verification accuracy of 99.38% on the famous face dataset: LFW (Labeled Faces in the Wild).

III. Person Re-identification

- Structure of Resnet-50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

The model of the feature extractor is RESNET 50. We apply the triplet loss propose in work [In Defense of the Triplet Loss for Person Re-Identification], which helps the model optimizes the features such that data points with same identity are closer to each other than those with different identity.

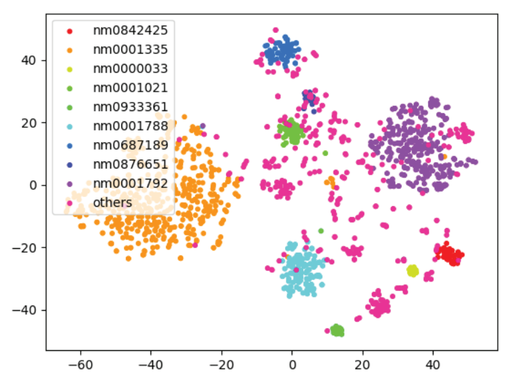
- Function of triplet loss

$$\mathcal{L}_{\text{BH}}(\theta; X) = \sum_{i=1}^{P}\sum_{a=1}^{K}\Big[m + \overbrace{\max_{p=1\ldots K} D\left(f_\theta(x_a^i), f_\theta(x_p^i)\right)}^{\text{hardest positive}} - \overbrace{\min_{\substack{j=1\ldots P\\ n=1\ldots K\\ j\neq i}} D\left(f_\theta(x_a^i), f_\theta(x_n^j)\right)}^{\text{hardest negative}}\Big]_+,$$

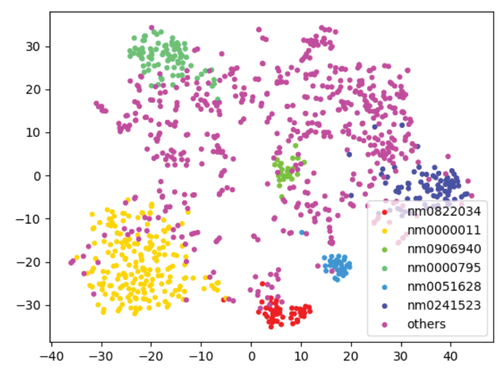
(all anchors)

- The model is trained using all the training dataset, excluding those labeled as "others".

train/tt0056869

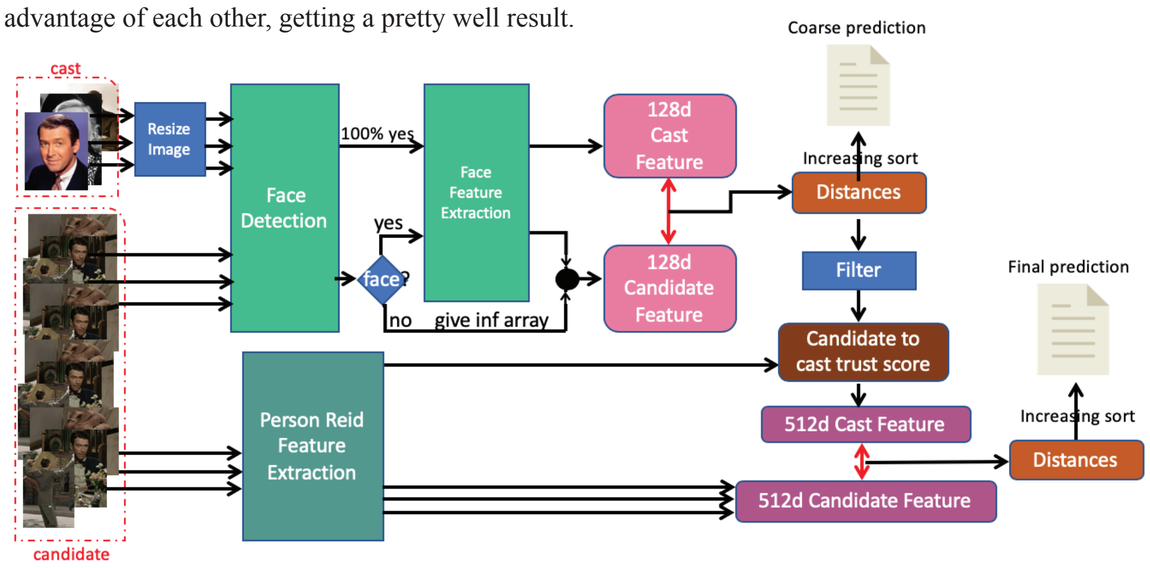


train/tt0027996



IV. Coarse to Fine System

Our coarse to fine cast recognition on movie stills pipline, combining part II and part III. Take the advantage of each other, getting a pretty well result.



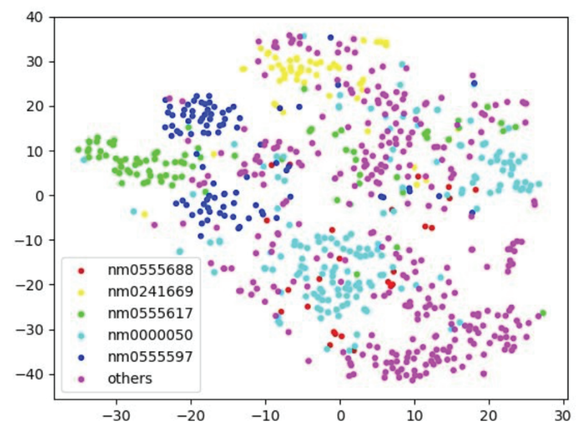
V. Experiment

Evaluation according to Mean Average Precision(mAP)

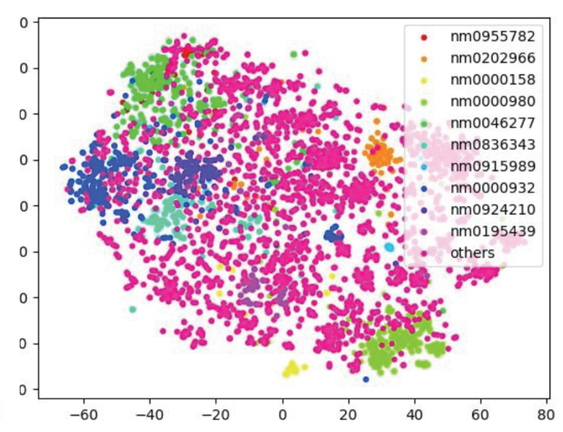
$$mAP = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{m_q}\sum_{k=1}^{n_q} P_q(k)rel_q(k)$$

where:
Q is the number of query cast
$\mathbf{m_q}$ is the number of the candidates with the same identity to the query
$\mathbf{n_q}$ is the number of all candidates in the movie
$\mathbf{P_q(k)}$ is the **precision** at rank *k* for the *q-th* query
$\mathbf{rel_q(k)}$ denotes the relevance of prediction *k* for the *q-th* query, it's 1 if the *k-th* prediction is correct and 0 otherwise

Visualization of movie "tt0023969"



Visualization of movie "tt1371111"



Ablation table (including val and test mAP)

model	val mAP	test mAP	notes
Dlib_inverse	0.3948	0.40796	inverse the relationship of cast and candidate
Dlib_force_location	0.4225	0.43808	force encoding by giving location of whole picture
Dlib_cnn_location	0.5010	0.51993	locating by cnn then encoding by that location
Dlib_allcast	0.4940	0.50801	aim to collect the most similar candiate to cast (inf mechanism)
SGBT_hybrid	0.5530	0.59749	hybrid state-of-art cluster and dlib pre-predicted label
SGBT_hybrid_pro	0.5631	0.	candidate number_of_times_to_upsample=0->1, num_jitters=1->2
SGBT_hybrid_resize	0.	0.	resize small image to large, large to small. cast&candidate num_jitters=1->5
_	0.	0.	_
_	0.	0.	_