VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Facial recognition from images (identification)

POVa - Počítačové vidění (v angličtině)

Vojtěch Vlach	xvlach22
Martin Kneslík	xknesl02
Zuzana Hrkľová	xhrklo00

1 Assignment

Prepare a demo application demonstrating facial recognition in good lighting conditions. Evaluate accuracy on you own data and on an existing dataset. Ideal approach is:

1. Detect faces using existing detector. Good choices are OpenCV, Dlib or DCGM/MTCNN.
2. Align the face based detected facial features (map to avarage face).
3. Extract face fingerprint using a convolutional neural network. You can start with some pretrained network or train or fine-tune your own - search Model Zoo for suitable network Model-Zoo.
4. Search database of faces.

2 Team responsibilities

- Zuzana Hrkľová - face detection and alignment, fine-tuning
- Martin Kneslík - datasets and image processing, demo application
- Vojtěch Vlach - fine-tuning and evaluation

3 Existing solutions

Facial recognition has undergone extensive development, with significant contributions from various datasets and models. The `FaceNet` model, introduced by Google [3], demonstrated the ability to map face images to a compact Euclidean space, where distances directly correspond to facial similarity. This approach laid the groundwork for many subsequent systems by emphasizing accurate embedding representation for faces.

VGGFace2 [1], another critical advancement, provides a large-scale dataset containing images with variations in pose and age. Trained on the InceptionResNetV1 architecture, it offers robust generalization across diverse facial features, enabling improved identification and verification. These foundational works have significantly influenced modern frameworks, including `facenet_pytorch`, utilized in our project for face detection and embedding extraction.

4 Task solution

Face detection. We compared several detectors including `OpenCV`, `Dlib` and `MTCNN` and decided to use `facenet_pytorch` in our implementation due to its high accuracy in detecting faces across various poses and lighting conditions. It performs face detection through `MTCNN`.

Face embedding extraction. We've chosen the facenet_pytorch library so far as it offers two pre-trained models: `VGGFace2` [1] and `CASIA-WebFace` [4]. Both models implement `InceptionResnetV1` [2] architecture varying only in training datasets.

Dataset. We will be using Large-scale CelebFaces Attributes (CelebA) Dataset. It contains 202599 aligned and cropped face images of 10177 identities. Each image is additionally annotated with class (identity), bounding box, landmarks positions (coordinates of left eye, right eye, nose, left mouth, right mouth) and 40 binary attributes (such as eyeglasses, mustache, wearing hat, etc.). Dataset is also partitioned into training, validation and testing set (1-162770 train, 162771-182637 val, 182638-202599 test).

Public project repository: `https://github.com/vlachvojta/POVa_face_identification`

Sample images from CelebA Dataset

5 Evaluation

Choosing correct measure proved to be a bit of a challenge, so we experimented with a few different metrics. In the core of everything, there is **cosine similarity** to compare two vectors directly, which ranges from -1 to 1.

For testing purposes we see face identification as **classification of faces** into classes/identities, typical classification metrics can be used like **accuracy** (ratio of correctly chosed classes) and **top-k accuracy** (ratio of correct class being in top k classes)

For validation during training we evaluated the **embedding space directly**, computing similarity of all embeddings with each other. Validation set of 1 000 images resulted in 1 000 x 999 similarity matrix after deleting the main diagonal. Next we created matrix of same shape with binary information whether the image pair is from the same class or not. Using these metrices we computed **accuracy** and **F1 meassure** with several thresholds (0.0, 0.4 and 0.8). And finaly **AUC (Area under the ROC curve)** as a way to consider **all thresholds at once**.

6 Experiments setup

In our experiments we started from two pretrained models on large datasets.

- VGGFace2 (Inception resnet using facenet_pytorch) pretrained specifically for face identification on VGGFace2 dataset.
- ResNet50 (torchvision) pretrained on more general image recognition tasks.

Both models have been trained on large datasets making them robust, We decided to find a weakness (e.g. model can't identify people with sunglasses) and focus on fine-tuning on this more specific problem while not reducing overall accuracy on the whole set. For this task we used CelebA dataset containing over 200 000 photos with not only identities but also 40 binary attributes (e.g. eyeglasses, smiling, mustache...)

First we had to find weaknesses, so we used **threshold accuracy** on image pairs of the same identity with different attribute value. Results can be seen in table 1. We've chosen **eyeglasses/sunglasses**, **lipstick** and **blurry photos** for our experiments.

Attribute	**Accuracy (0.6)**	**Accuracy (0.7)**	**Accuracy (0.8)**	**Number of pairs**
eyeglasses/sunglasses	**0.47**	**0.21**	**0.04**	14512
blurry photo	0.50	0.26	0.07	16870
wearing lipstick	0.58	0.31	0.09	**34745**
wearing hat	0.56	0.33	0.11	14279
heavy makeup	0.63	0.37	0.11	44659
blond hair	0.63	0.37	0.12	26260
...	...	...	...	...

Table 1: Measure of accuracy based on paired attributes (image with/without attribute)

	trn images	**trn classes**	**ResNet50**		**VGGFace2**	
			AUC balanced	AUC orig	AUC balanced	AUC orig
orig data	**162770**	**8192**	–	0.71	–	0.72
eyeglasses	13306	2598	**0.87**	**0.86**	**0.85**	**0.84**
blurry photo	16456	3435	0.81	0.82	0.63	0.66
wearing lipstick	28654	3578	0.72	0.77	0.65	0.73

Table 2: Fine-tuning experiments with balanced attributes. AUC (Area under ROC curve) consideres all possible thresholds.

For fine-tuning on a specific attribute we selected subset of both training and validation sets in a way that assured a same number of True and False attribute photos for every identity. This partly reduced training dataset size (details in table 2). To ensure the overall accuracy doesn't decrease, we validated training on both attribute balanced dataset as well as the original validation dataset.

Preprocessing of images. Every image is ran though face detection (dataset has too big space around faces), scaled to 160x160 pixels and normalized. **The preprocessing for VGGFace2** model is taken from its documentation and images are normalized to interval [-1, 1] and the square resolution is achieved by squishing the face detection directly, resulting in vertically squished face from our point of view, but more dense information for the model. **Preprocessing images for ResNet50** takes detected face and adds padding around to make a square, resulting in a face without deformations. Image is also normalized using ImageNet normalization to these values for color channels. Mean: (0.485, 0.456, 0.406). Std: (0.229, 0.224, 0.225).

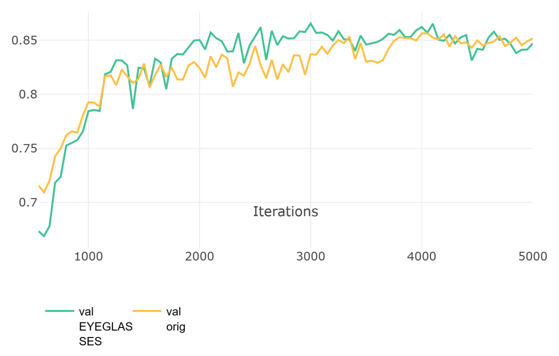
For each selected attribute we ran training for 5 000 iterations and both models with `ArcFaceLoss` and `TripletMarginMiner` for mining positive and negative examples. We used batch size 16 and trained on GeForce GTX TITAN X. Learning rate has been set to 2e-4 and weight decay to 0.01. Classic Adam optimization.

7 Experiment results

Results are shown in table 2. Considering initial AUC orig to be around 0.7 in every experiment, it is easily said what effect the experiment had on overall accuracy. While training on balanced blurry/not-blurry photos seems to decrease both accuracies for VGGFace2, ResNet50 seems to get only better. The best result for both models is considered to be experiment with eyeglasses increasing not only balanced AUC but even the AUC on the original validation set.

Meassured AUC behaved differently for two models as visible in Figure 7. Training ResNet proved to be more straight-forward, as ResNet50 is pretrained on general image recognition tasks. VGGFace2 on the other hand needed around 2 000 iterations to stabilize because it was trained on highly specific data and our preprocessing might not be entirely the same.

To add a different view on our trained models, we measured classification on the test set with 250 classes. See table 3. The VGGFace2 model had pretty good accuracy already but our training seems to decrease it, while ResNet model was not ideal and our training upgraded it on the same level as VGGFace2 model.



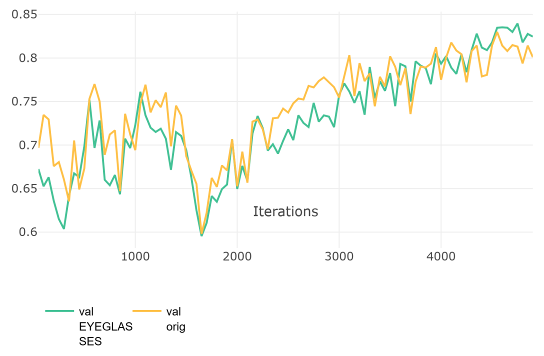


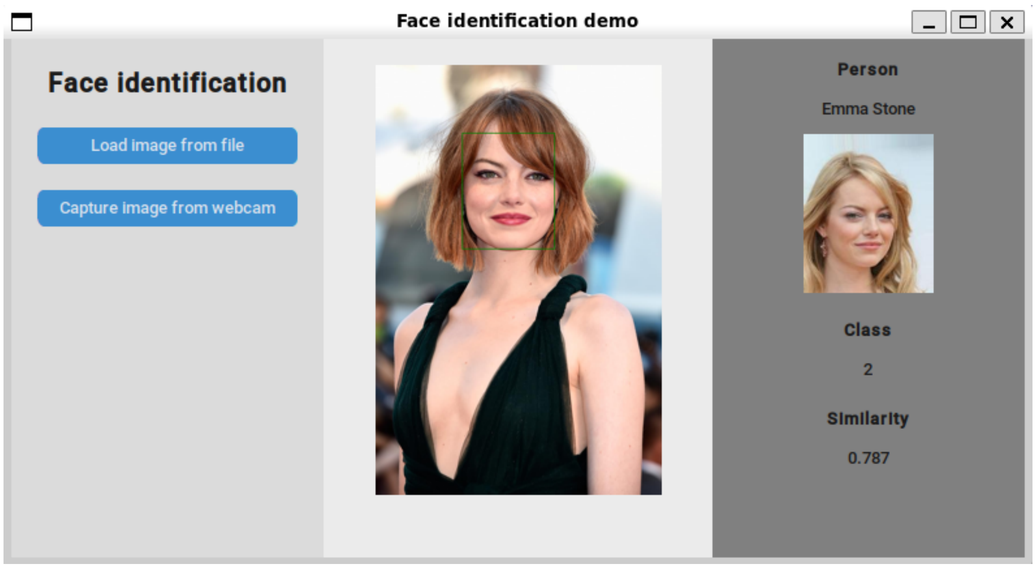
Figure 1: AUC (Area under ROC Curve) for fine-tuning Resnet50 (left) and VGGFace2 (right) on eyeglasses. Resnet50 seems to pick up the training progress right away, whereas VGGFace2 needs more time to stabilize the training process.

Model	**Top-1**	**Top-2**	**Top-3**
Resnet (pretrained)	.68	.76	.80
VGGFace2 (pretrained)	**.98**	.99	.99
VGGFace2 eyeglasses	.64	.73	.78
Resnet eyeglasses	**.94**	.97	.98

Table 3: Classification accuracy. Top-N meaning correct class was in top-N ordered by similarity (top-1 means correct class had biggest similarity)

8 Demo application

We have created a simple demo application that identifies a person from an image (which is loaded from a file or a webcam). It identifies persons from a "database" - reference images of the person in a folder. After the image is loaded, it identifies the person and shows the name, class, and a reference photo.



Person

Emma Stone

Class

2

Similarity

0.787

Figure 2: Demo application

9 Conclusion

The goal of this project was to develop a facial recognition system from images. To achieve this, we implemented and evaluated a system utilizing two pre-trained models (VGGFace2 and ResNet50) alongside fine-tuning techniques. Our experiments demonstrated the effectiveness of attribute-focused training in improving recognition accuracy under challenging conditions such as sunglasses, lipstick, and blurry images. The results confirmed the robustness of the selected models and methodologies while highlighting areas for further optimization. The resulting demo application provides a practical example of face identification, showcasing its potential for user-friendly face recognition.

References

[1] Cao, Q., Shen, L., Xie, W., Parkhi, O. M. and Zisserman, A. VGGFace2: A Dataset for Recognising Faces across Pose and Age. In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. 2018, p. 67–74. DOI: 10.1109/FG.2018.00020.

[2] Peng, S., Huang, H., Chen, W., Zhang, L. and Fang, W. More trainable inception-ResNet for face recognition. *Neurocomputing*. 2020, vol. 411, p. 9–19. DOI: https://doi.org/10.1016/j.neucom.2020.05.022. ISSN 0925-2312. Available at: `https://www.sciencedirect.com/science/article/pii/S0925231220308572`.

[3] Schroff, F., Kalenichenko, D. and Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2015, p. 815–823. DOI: 10.1109/cvpr.2015.7298682. Available at: `http://dx.doi.org/10.1109/CVPR.2015.7298682`.

[4] Yi, D., Lei, Z., Liao, S. and Li, S. Z. Learning face representation from scratch. *ArXiv preprint arXiv:1411.7923*. 2014.