



# Applied Deep Learning

Project Presentation

Zhi Zheng (zz2560)

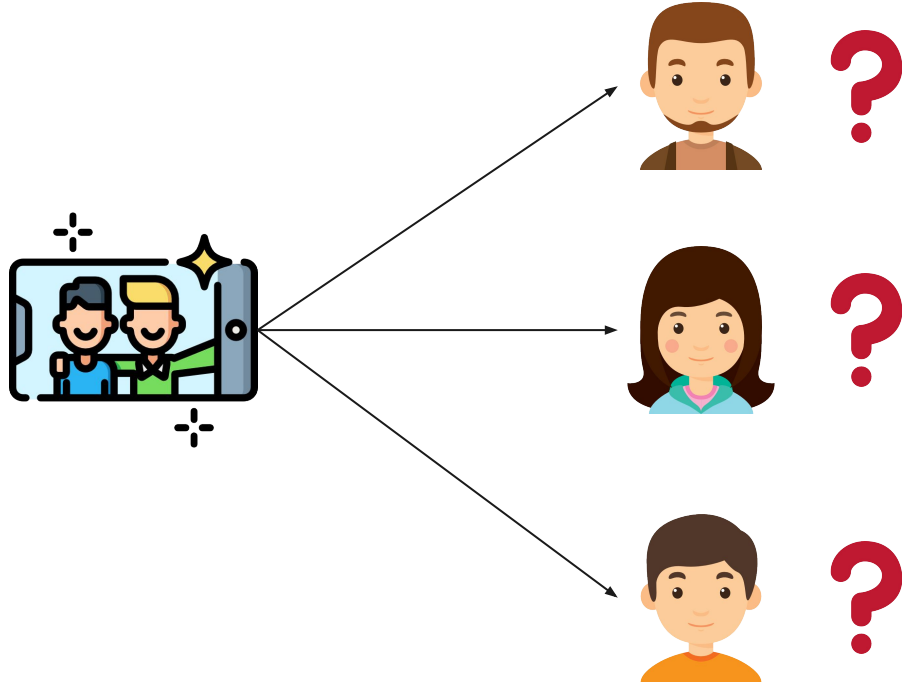
Da An (da2841)

# Content



1. Problem Description
2. Related Works
3. Our Solution
  - a. Workflow
  - b. Model training
4. Result
  - a. Training result
  - b. A frontend only website for face clustering
5. Demo
6. Limitations

# Problem Description



*Who is in the photo?*

# Related Works



1. Schroff, F., Kalenichenko, D. and Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
2. Howard, A, Zhe, M. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications
3. Tensorflow.js

# Our Solution



An end-to-end javascript photo clustering service.

Use Tensorflow.js to make the inference process run on local machine to maximize privacy

Use MobileNet as primary model to reduce the model size

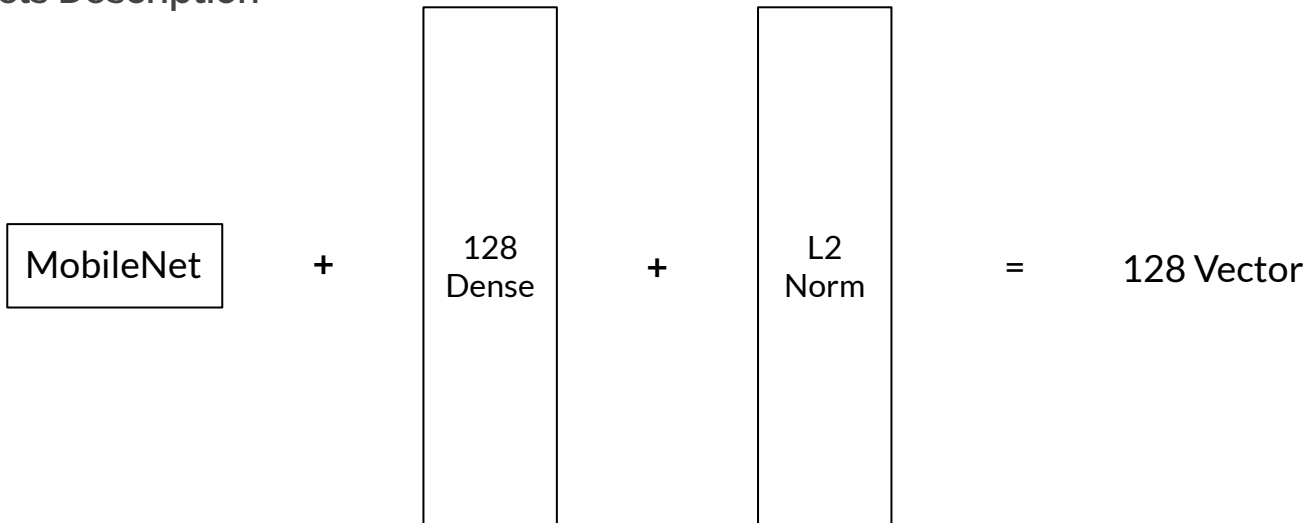
# Our Solution



Workflow

# Our Solution

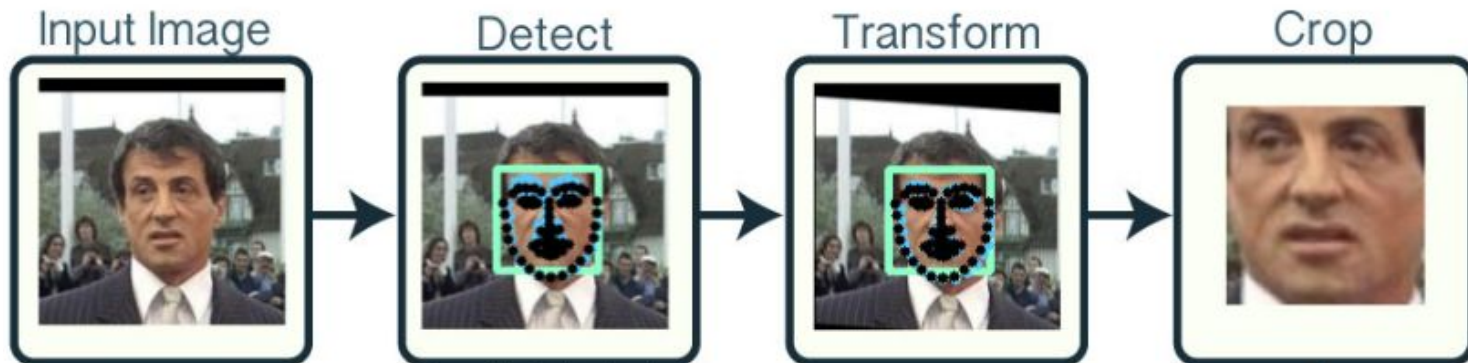
Models Description



# Our Solution

## Dataset + Preprocessing

1. Dataset: CASIA\_Webface (10000+ person, average 50 photos/person)
2. Detect face in the photo
3. Align face in the photo
4. Normalization



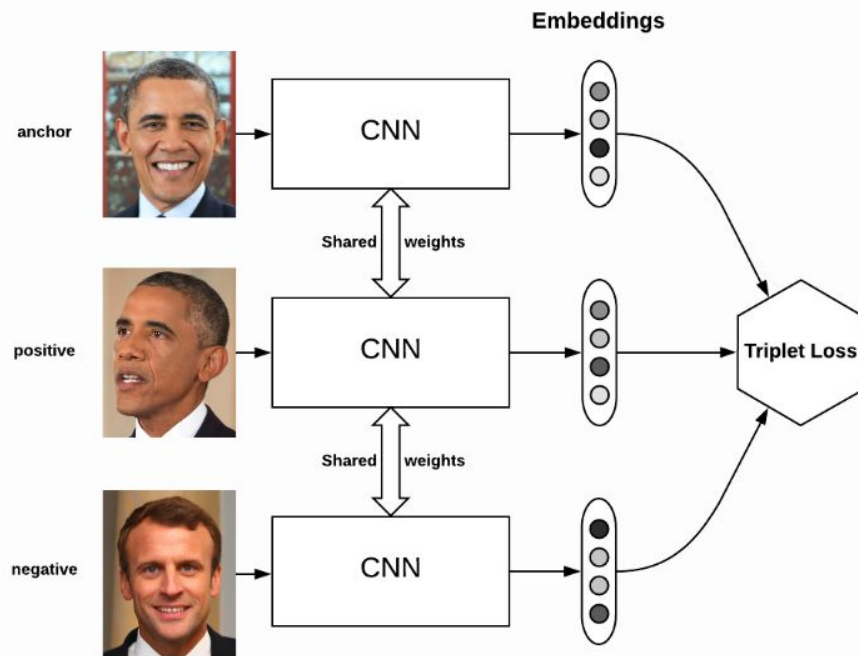


# Our Solution

## Loss function

1. **Loss: Triplet Loss:** give three images, one anchor, one positive and one negative, the goal is to maximize the distance between (positive - anchor) and (negative - anchor)
  - a. Batch-Hard
  - b. Batch-All

$$\mathcal{L} = \max(d(a, p) - d(a, n) + \text{margin}, 0)$$

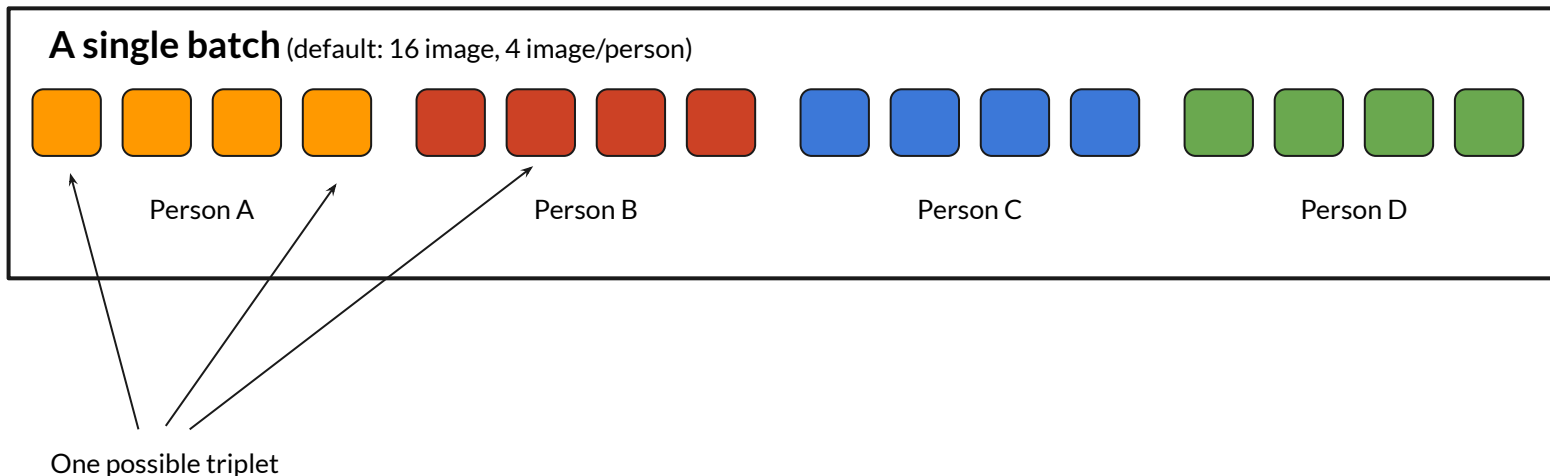


Triplet loss on two positive faces (Obama) and one negative face (Macron)

# Our Solution

## Data Generator

1. To ensure that we can always find all three images in a single batch, we define a data generator for training.
2. Trained on 4415 person with 30 photos each person, total: 132, 450 photos



# Our Solution



## Evaluation

1. Evaluation Dataset: "Label Face in the Wild"
2. Accuracy: 50.1%... (trained on 130, 000 photos)
3. Model size: 14MB (compare to 100MB+ in original Facenet paper)

#training images	VAL
2,600,000	76.3%
26,000,000	85.1%
52,000,000	85.1%
260,000,000	86.2%

Source: facenet paper

# Our Solution + Demo



Tensorflow.js + Pretrained model

We build a fully Javascript service using pretrained model.