Face Recognition with Sub-sampled Images

Presentation link: https://youtu.be/C-hF4lhhQzg
Chung Chun Wan (cwan@), Yuxuan Wang (ywthree@)
CS230 Deep Learning

Motivation

Given a particular sensor resolution and face recognition block, there is a trade-off between the camera's field of view and the working distance. In this project, we explore a deep learning-based approach such that given a particular face recognition block, e.g. FaceNet, we can significantly reduce the input image size (e.g. from NxN to N/8xN/8) while matching as much as possible the original accuracy.

Dataset

Labeled face in the wild (LFW):

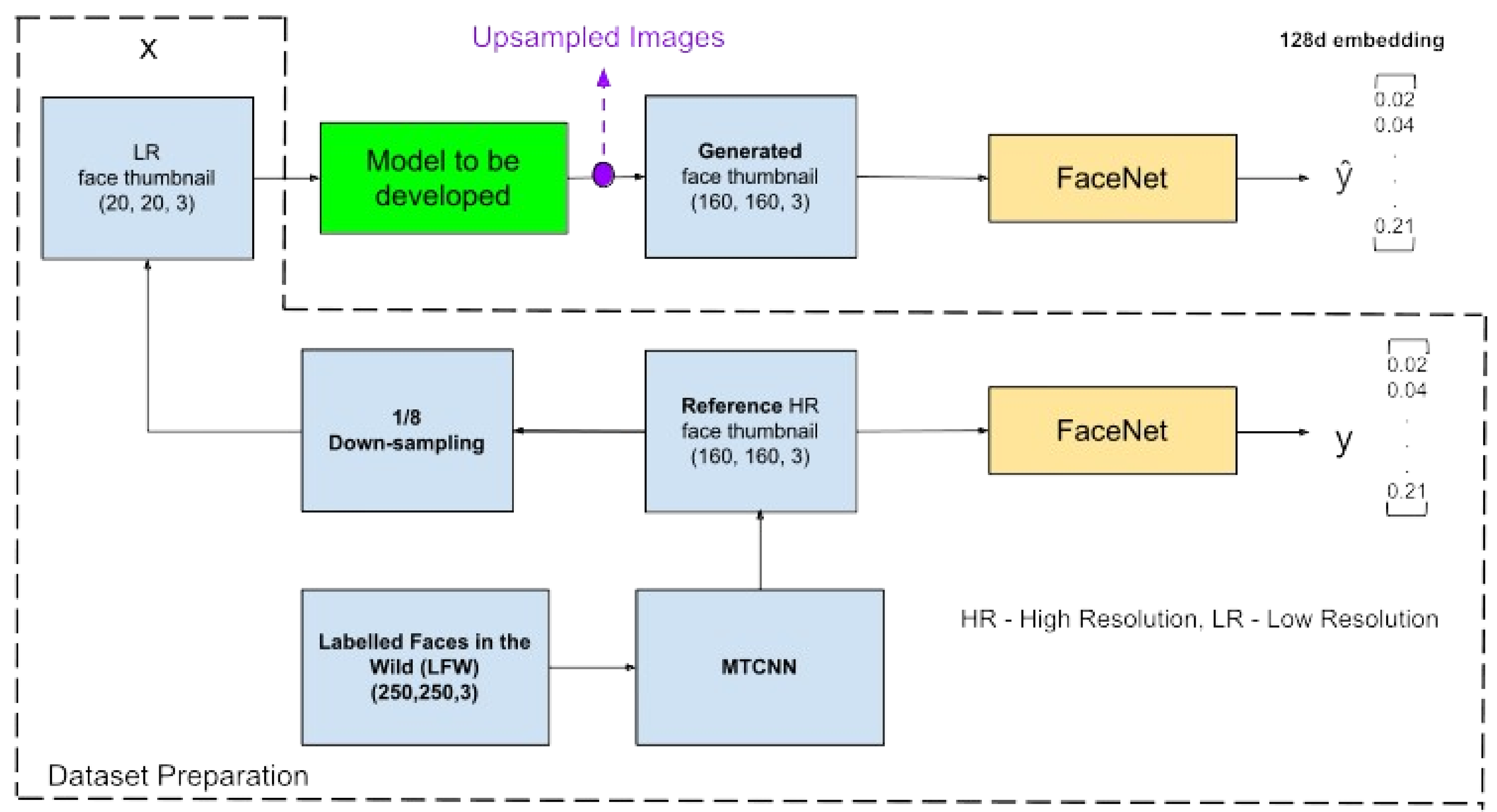
- 13,233 RGB images with a size of (250, 250)
- Training set: 11,910 images – 90%
- Development set: 972 images – 7.35%
- Test set: 351 images - 2.65%

Data preparation:

- Crop face bounding boxes out of images with an open-source face detector – MTCNN. All boxes are resized to (160, 160) with PIL.image.resize()
- Prepare training, dev/test datasets:
  - X, **low-resolution face images**, by down-sampling the bounding boxes to (20, 20) with PIL.image.resize()
  - Y, **128-dim embedding vectors**, by feeding high-resolution bounding boxes (160, 160) to an open-source FaceNet
- Prepare data for performance comparison:
  - high-resolution images X' with interpolation-based methods – nearest, bilinear and bicubic, from X
  - embedding vectors for X'

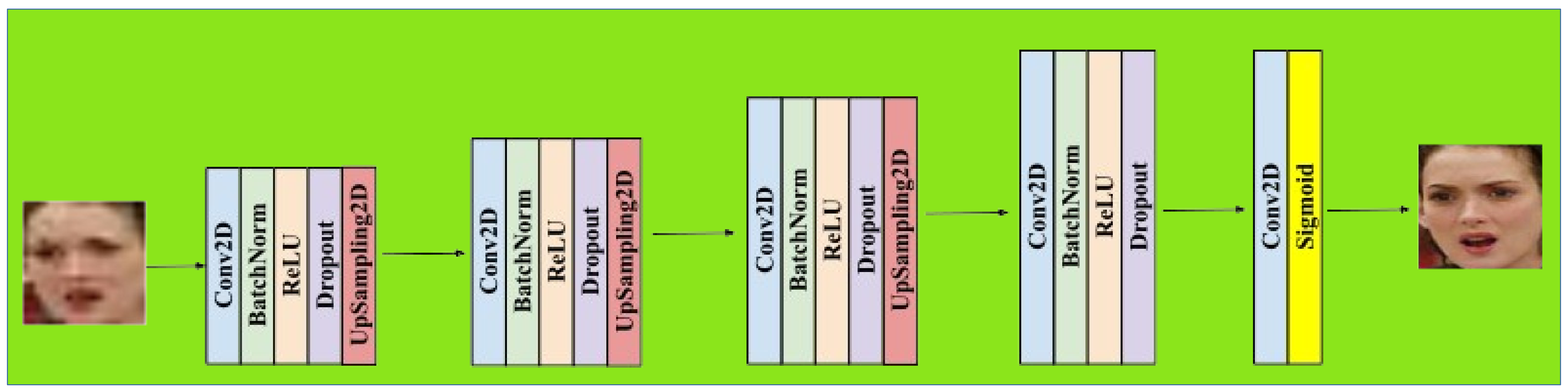
Model and Results

Training pipeline:



$$\text{Cost Function } J = \frac{1}{n}\sum_{i=1}^{n} \|y - \hat{y}\|^2$$

Model architecture:



Filters: 256	Filters: 128	Filters: 64	Filters: 32	Filters: 3
Size: 3x3	Size: 3x3	Size: 3x3	Size: 3x3	Size: 3x3
Padding: same	Padding: same	Padding: same	Padding: same	Padding: same
Stride: 1	Stride: 1	Stride: 1	Stride: 1	Stride: 1

Performance:

# of samples that our model performs better or worse than nearest, bilinear and bicubic interpolations

	NEAREST	BILINEAR	BICUBIC
Better	10991	6999	6691
Worse	919	4911	5219

Training Set (Total Images = 11910)

	NEAREST	BILINEAR	BICUBIC
Better	861	496	467
Worse	111	476	505

Dev Set (Total Images = 972)

	NEAREST	BILINEAR	BICUBIC
Better	302	166	167
Worse	49	185	184

Test Set (Total Images = 351)

Positive and negative examples. Top row - our model performs better. Bottom row – Bicubic interpolation performs better. From left to right, input image, ground truth HR image, bicubic upsampled image, upsampled image with our model.

Conclusions

- Developed a deep learning model that up-samples aggressively sub-sampled images (⅛ the resolutions in both image dimensions)
- The model outperforms both Nearest and Bilinear interpolations but slightly under-perform Bicubic interpolation

Future Work

To further improve the model performance, need to try

- Bigger filters (i.e., 5x5) since the bicubic interpolation explores information from 4x4 neighboring pixels
- Different architectures, such as GAN framework or DenseNet
- Different loss functions, such as L1