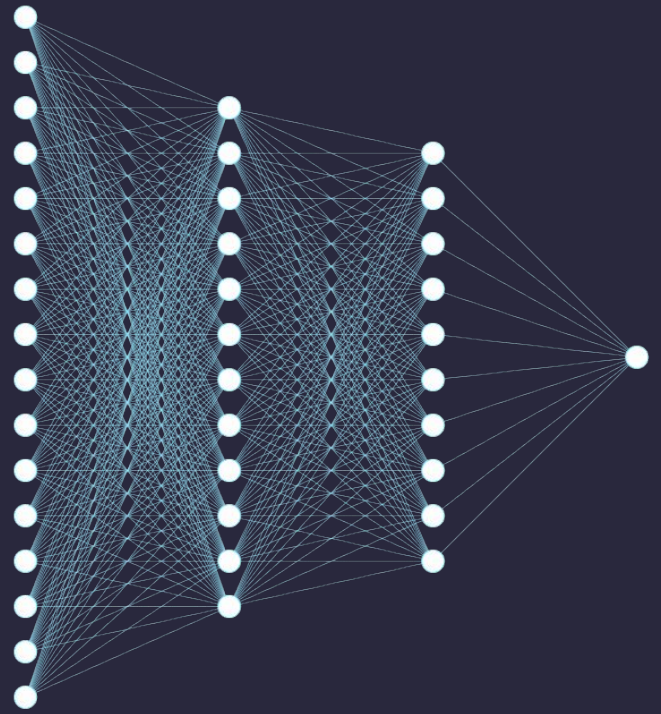
Flower Recognition: Dealing with Less Data via Few-Shot Learning

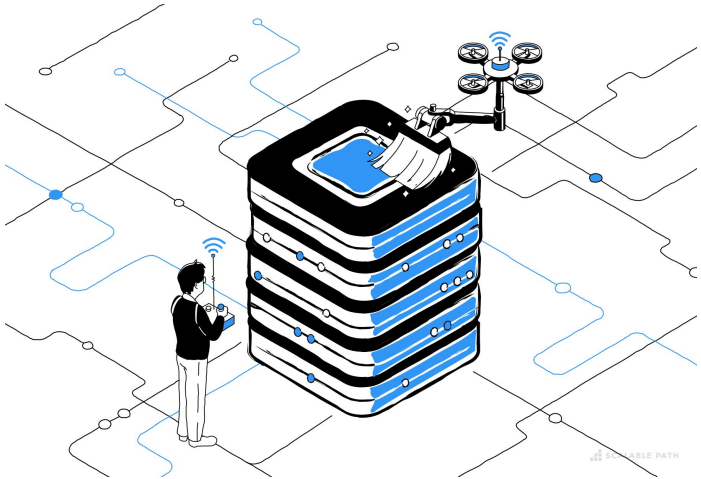
George Rahul
Chopra Dhruv
Agarwala Pratham



/TABLE OF CONTENTS

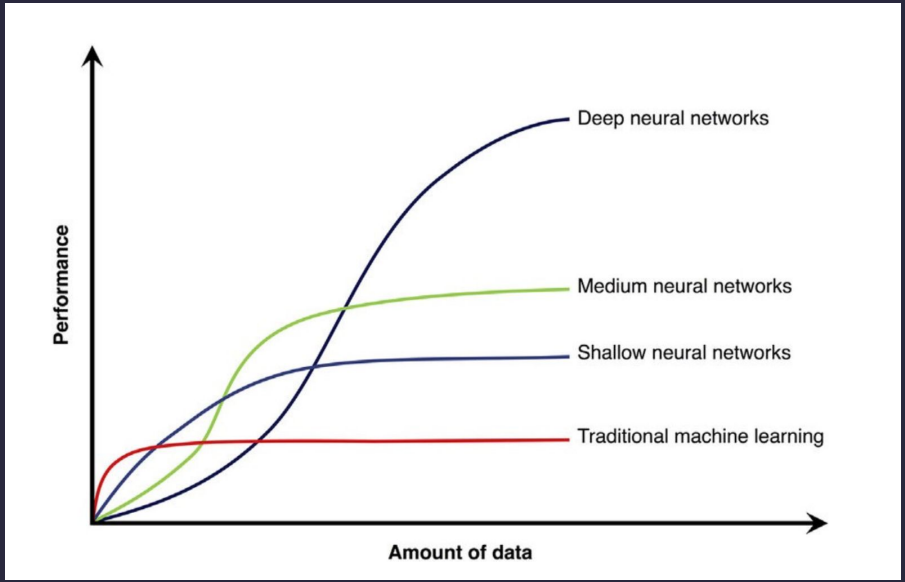
/01

MOTIVATION AND DATA AUGMENTATION



THE FELINE FALLACY

THE HUNGER OF NNs



Can you predict with insufficient training data?

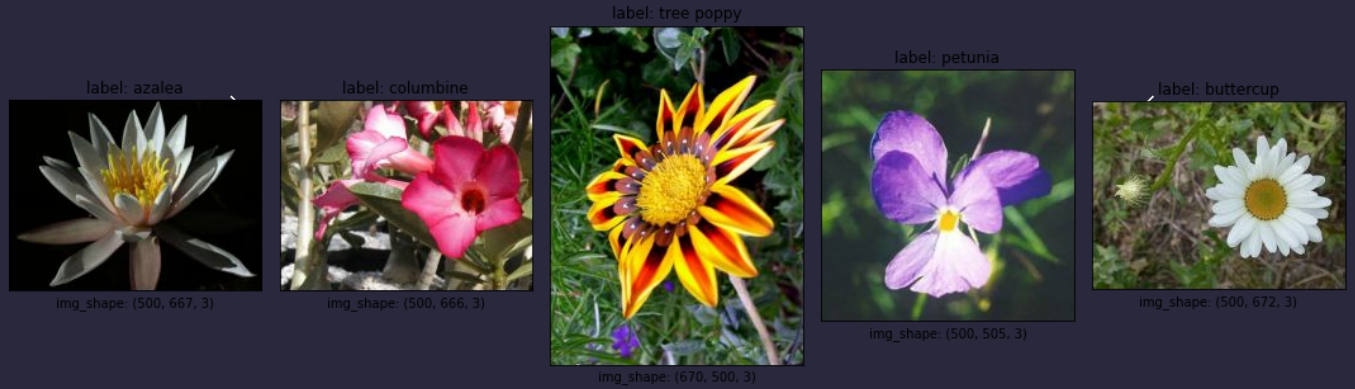
Human beings seem to be very good at recognising faces, cats, flowers etc. with only a few examples

LITERATURE REVIEW

- Deep transfer learning (Tan et al., 2018).
- First Siamese Network for signature verification (Bromley et al., 1993).
- Face Verification using Siamese Networks, paper by Facebook (Taigman et al., 2014).
- Triplet Loss function, paper by Google (Schroff et al., 2015).

<DATASET INFO!>

The Oxford Flowers 102 dataset is a consistent of 102 flower categories commonly occurring in the United Kingdom. Each class consists of between 40 and 258 images. The images have large scale, pose and light variations.



<DATASET INFO!>

The dataset is divided into a training set, a validation set and a test set.

The training set and validation set each consist of 10 images per class (totalling 1020 images each).

The test set consists of the remaining 6149 images (minimum 20 per class).

DATA AUGMENTATION

We are performing following types of transformations

RESIZING

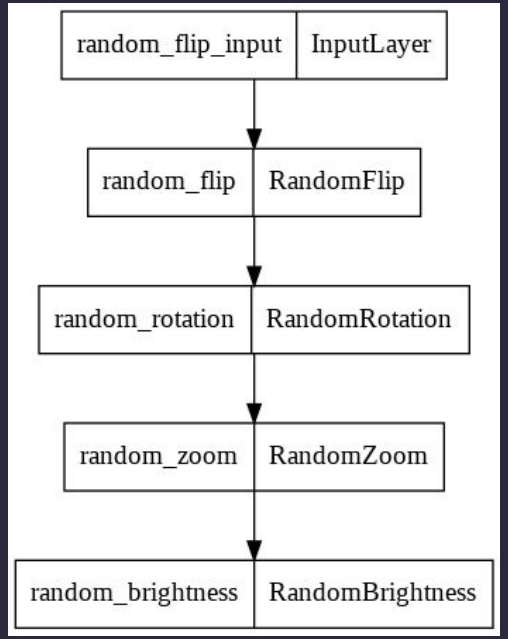
RESCALING

FLIPPING

ROTATION

ZOOM/CROP

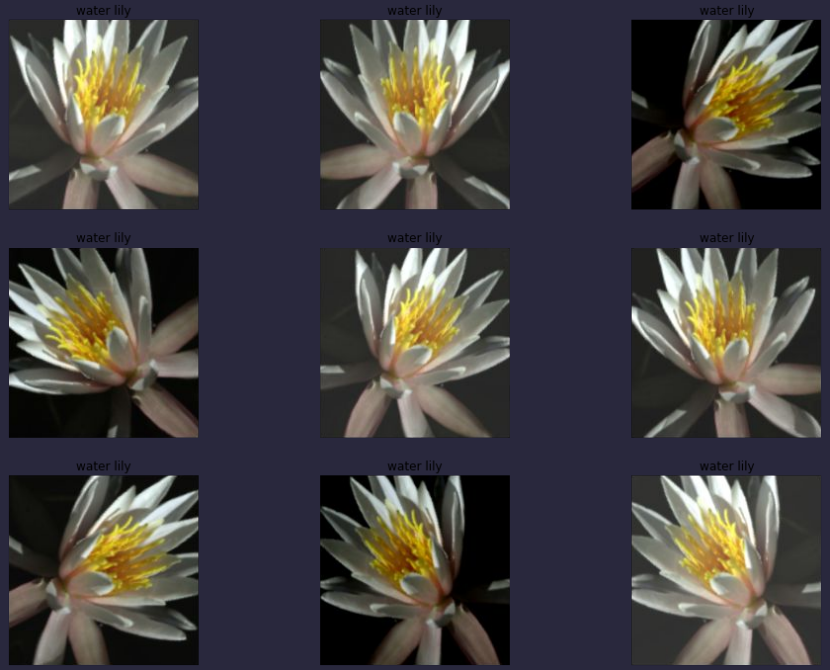
BRIGHTNESS



RESULTS AFTER AUGMENTATION

We used the repeat function of Tensorflow Dataset to repeat Our Random Image Augmentation on the Dataset.

Now we have over 5000 training images.



/02

TRANSFER LEARNING

Using Pre-trained Networks and fine-tune to apply on our dataset

TRANSFER LEARNING

- First step is add our own classification layer and train the network with keeping the pretrained network freezed.
- The second step is fine-tuning where we unfreeze a some layers and fit the model using smaller learning rate($1e^{-5}$).

DIFFERENT ARCHITECTURES

/1 INCEPTION.V3

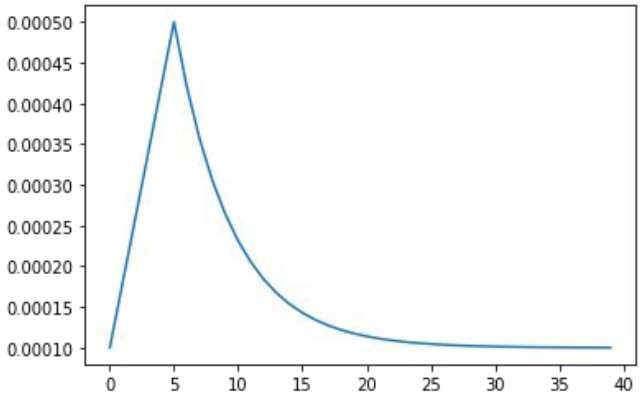
/2 DENSENET 201

/3 BigTransfer (BiT)

LEARNING CURVE

We are using Adam optimizer, along with a Learning Rate Scheduler which decays the learning rate after a certain number of epochs.

The learning rate then decays exponentially for following epochs.



TUNING MODEL ARCHITECTURE

1. **Increasing Depth of the Network**
2. **Adding Batch Normalization Layers**
3. **Dropouts**

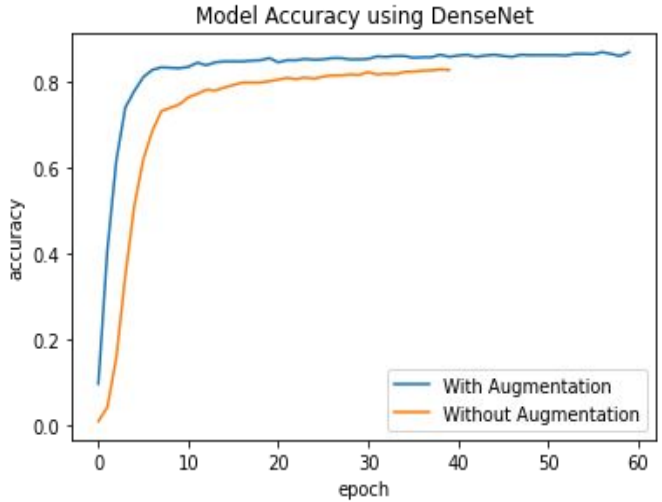
We perform Bayesian Optimization Tuning using Keras tuner to obtain best parameters.

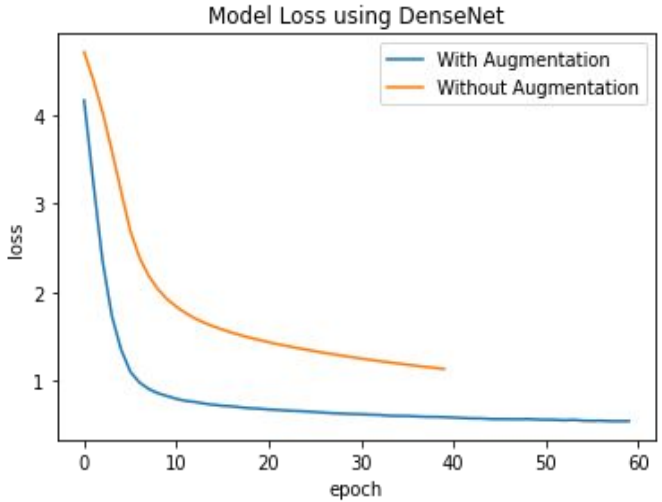
/MODEL RESULTS ON TEST DATA

MODEL	ACCURACY	LOSS
Altered BiT	0.985	0.0817
BiT + Softmax	0.982	0.1011
Altered InceptionV3	0.75	1.03
DenseNet 201 + Softmax	0.90	0.4231

EFFECT OF AUGMENTATION ON TRAINING

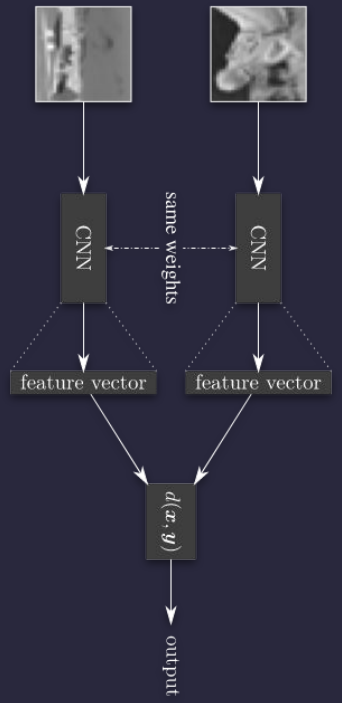
Below graphs represent validation loss across epochs trained for the DenseNet Model with and without Data Augmentation.





/03

FEW-SHOT LEARNING: SIAMESE NETWORK

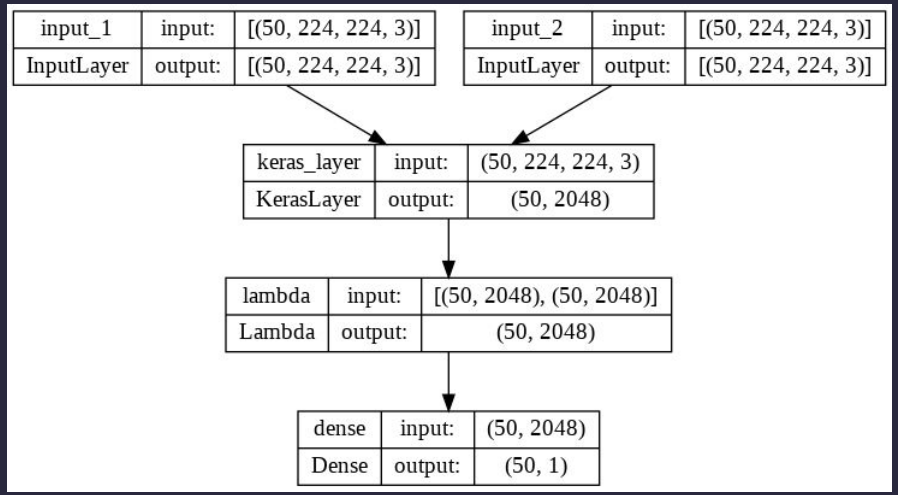


ARCHITECTURE

We used the BiG Transformer Feature Extractor to encode the images.

And a custom L2 layer to calculate the distance between the encodings.

Network was trained with binary cross entropy loss.



PREPARING THE DATA

We paired up the images within the batches to reduce complexity.

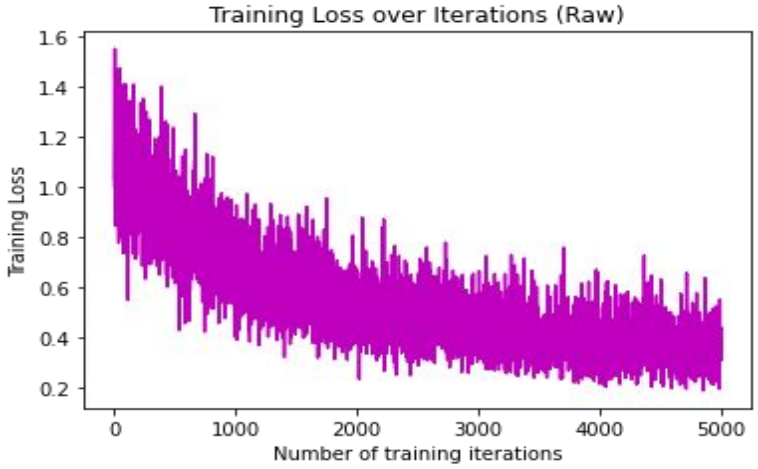
Each batch had a 1:1 ratio of similar and dissimilar pairs.

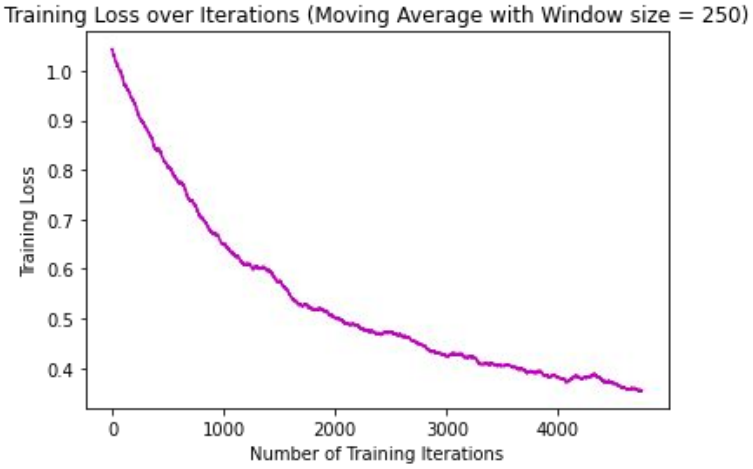
We used batch size 50 and pairs were formed by random sampling.

Images courtesy of Oxford Flower 102

TRAINING THE MODEL

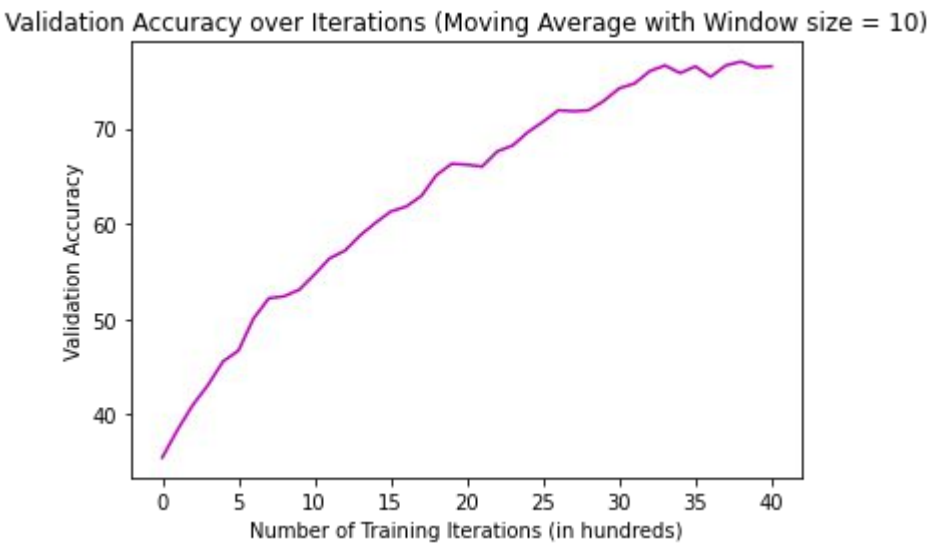
```
# Remember to compile the model!
optimizer = SGD(learning_rate = 0.0001)
model.compile(loss="binary_crossentropy", optimizer=optimizer)
```





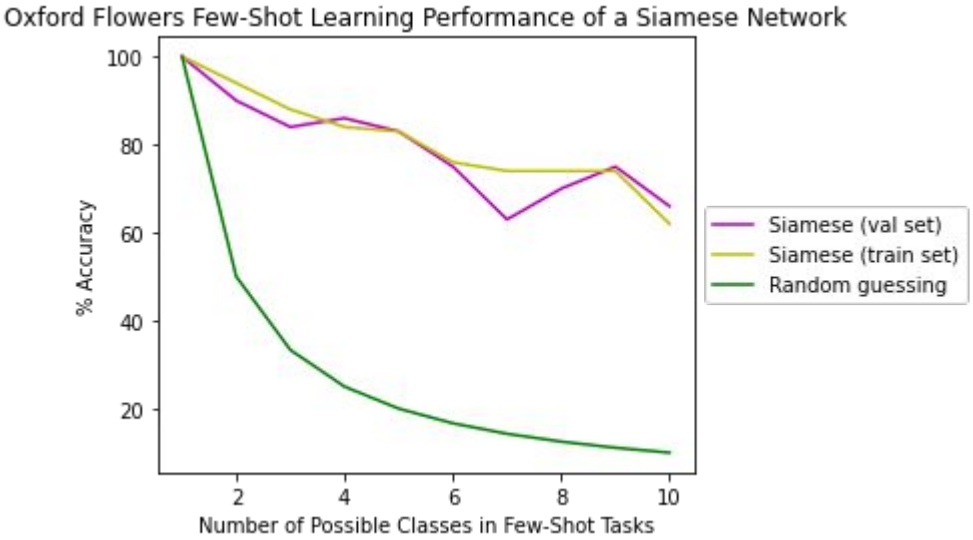
VALIDATING THE MODEL

```
# N_way = N; n_val = k (num trials)
N_way = 5 # how many classes for testing one-shot tasks
n_val = 100 # how many one-shot tasks to validate on
```



Images courtesy of Oxford Flower 102

VARYING THE NUMBER OF WAYS

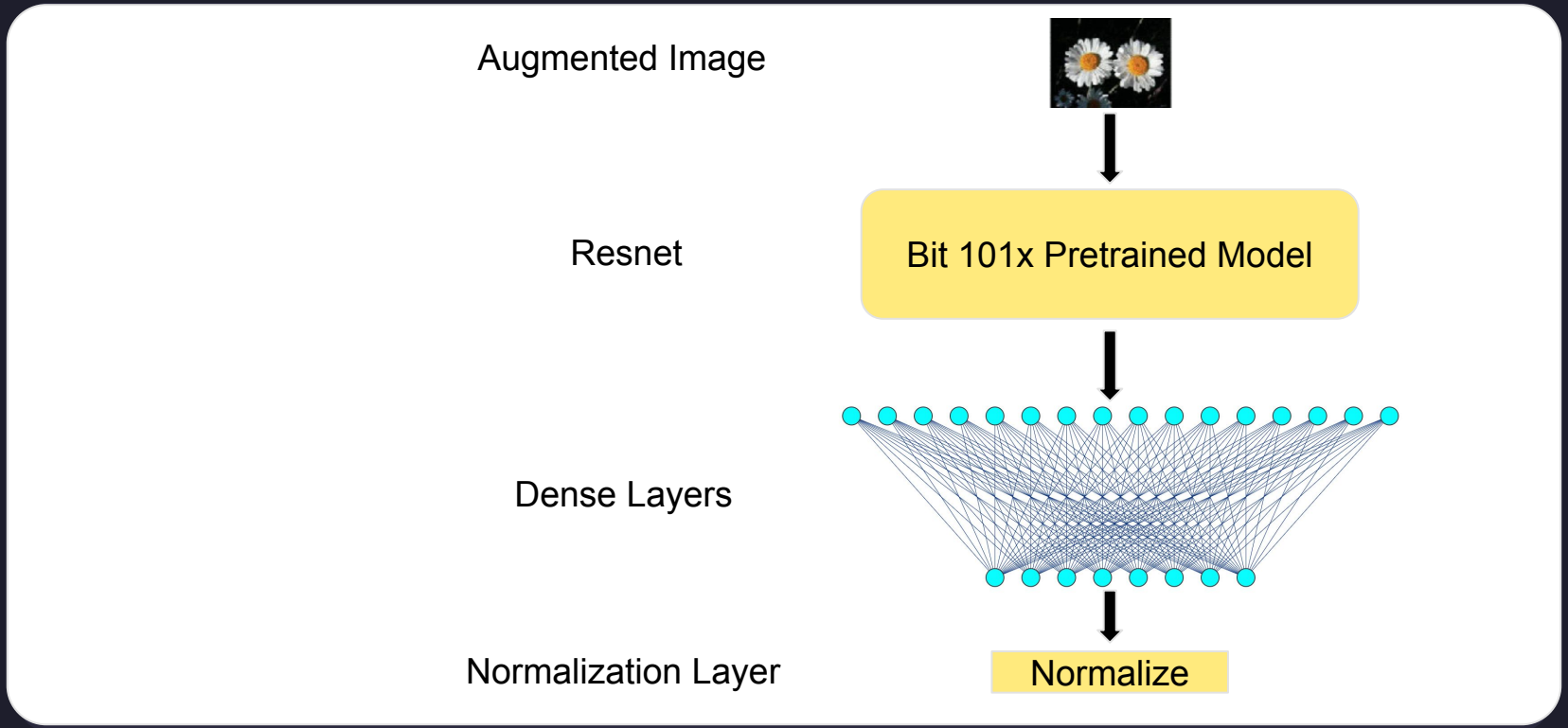


/04

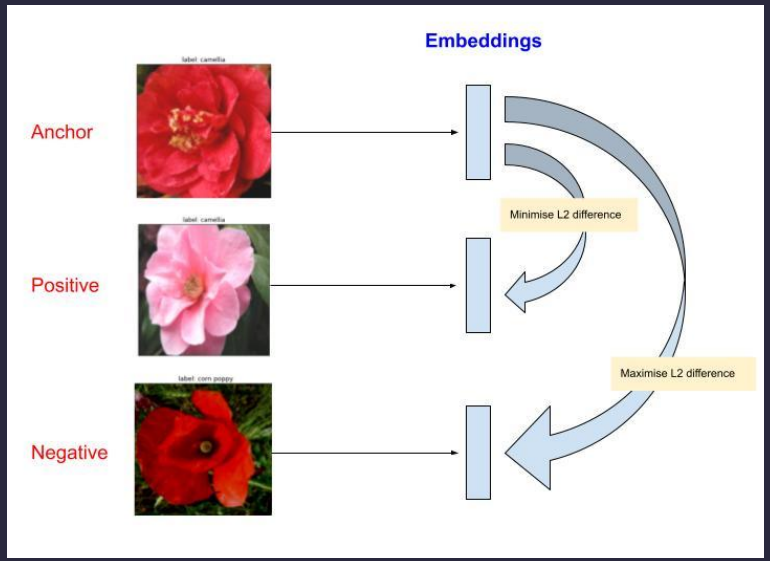
FEW-SHOT LEARNING: TRIPLET LOSS

Image courtesy Schroff et al., 2017 (FaceNet)

/Deep Architecture



/Some Terminologies



- **Anchor** — Given Image
- **Positive** — Category = Anchor Category
- **Negative** — Category =≠ Anchor Category

Images courtesy of Oxford Flower 102

/Improving the Model Loss

$$L2(rose_1, rose_2) < L2(rose_1, marigold_1)$$

Difference could be very small

$$L2(Anchor, Positive) < L2(Anchor, Negative) - margin$$

Ensure Loss is Positive

$$TripletLoss(A, P, N) = max(0, L2(A, P) - L2(A, N) + margin)$$

Sampling Triplets

& Semi-Hard Loss

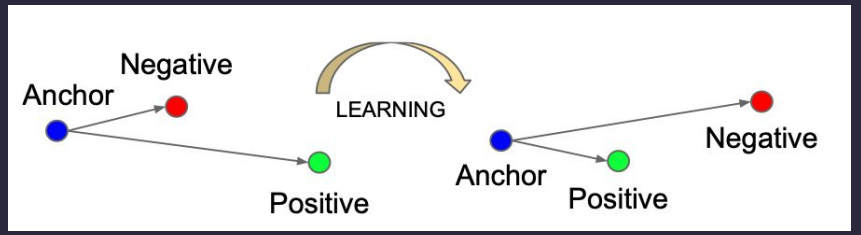


Image courtesy Schroff et al., 2017 (FaceNet)

/MODEL RESULTS ON TEST DATA

	F1-Score	Recall	Precision
Non-weighted Avg	0.99	0.99	0.99
Weighted Avg	0.99	0.99	0.99

Achieved an accuracy of 0.9882907789884534

/VISUALIZING THE EMBEDDINGS USING PCA

https://projector.tensorflow.org

/DEPLOYED APPLICATION FOR DEMONSTRATION

https://504910d1ba35053b.gradio.app/

THANK YOU