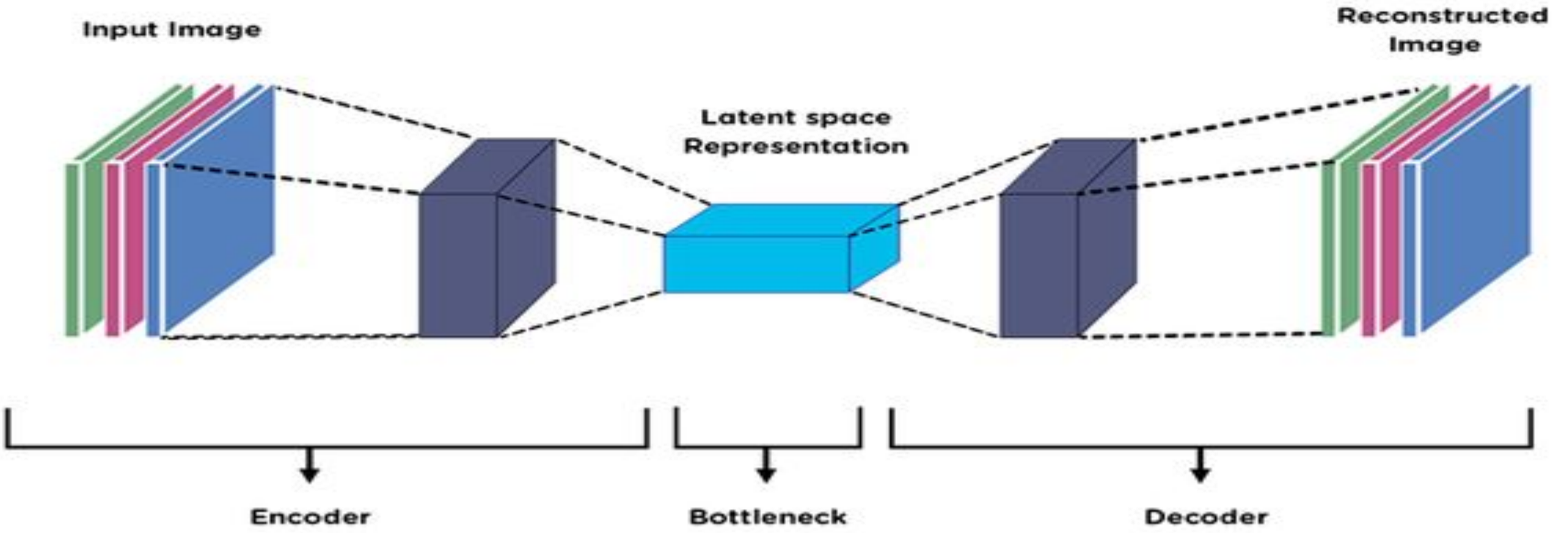
Comm system optimization assisted by deep-learning algorithms

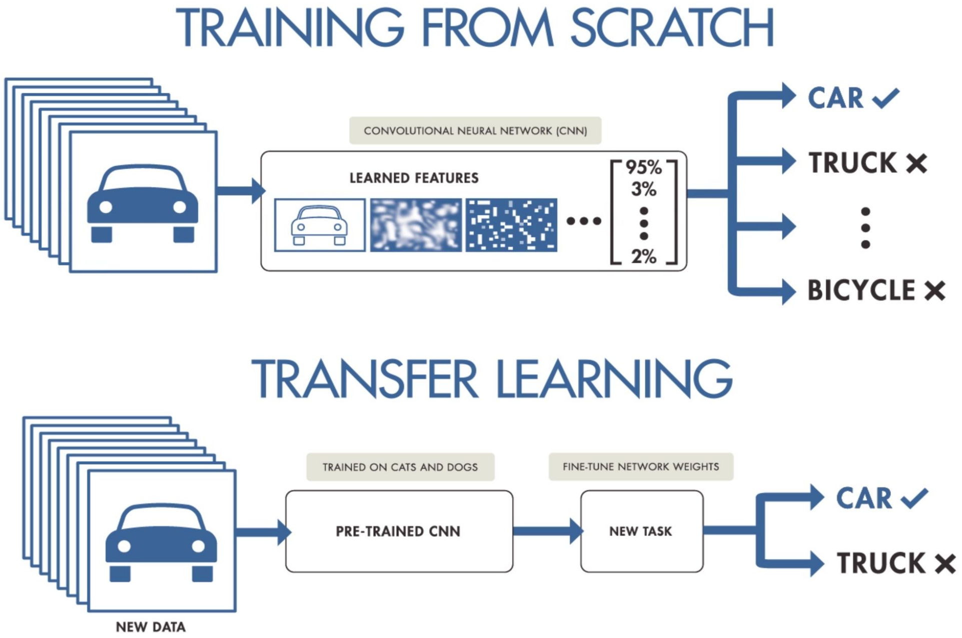
TOOLS & AlGORITHMS

I. AUTOENCODERS



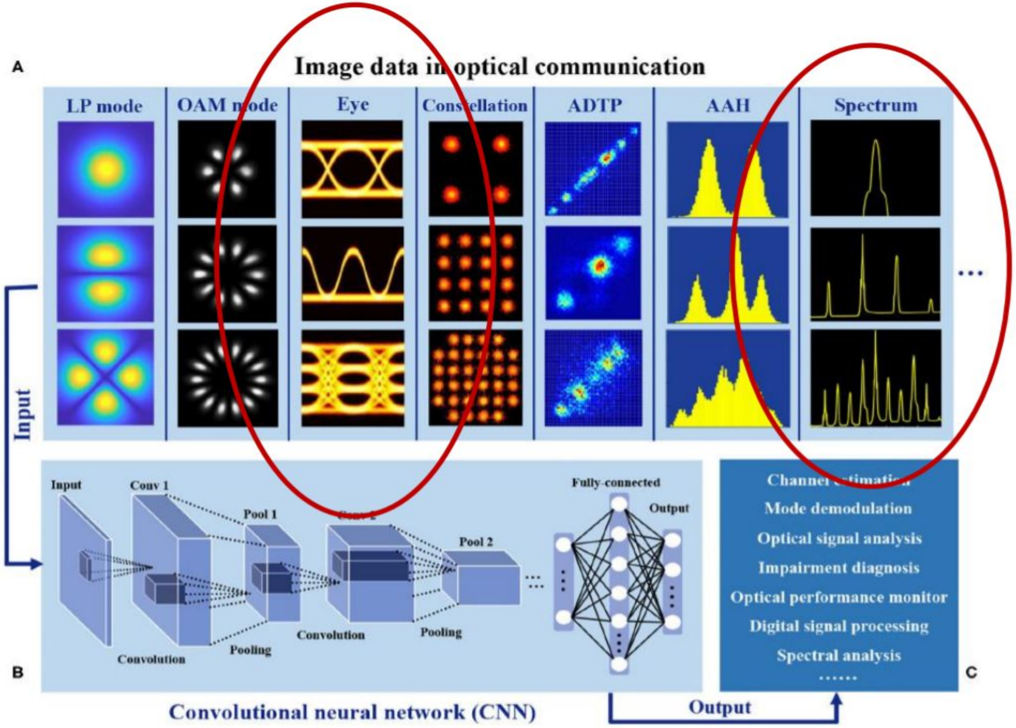
TOOLS & AlGORITHMS

II. TRANSFER LEARNING



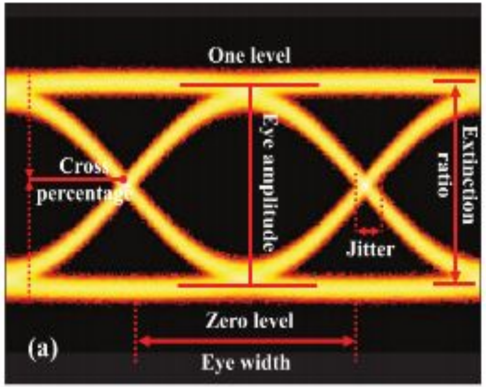
PROBLEM STATEMENT

Using eye diagrams for evaluating system performance and gaining insight into the nature of channel imperfections that can lead to errors using deep learning algorithms

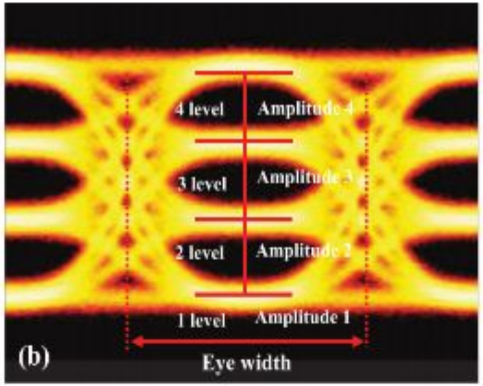


INTO THE EYE DIAGRAM

- Characteristic parameters
- Overall signal performance indicators
- Fiber length estimation
- Device Imperfections



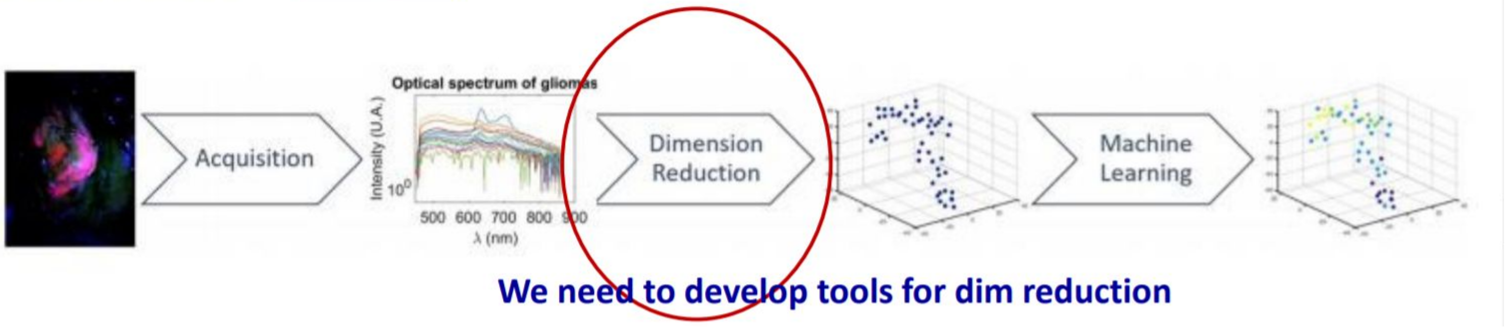
a)OOK



b)PAM4

CHALLENGES

I. DIMENSIONALITY REDUCTION

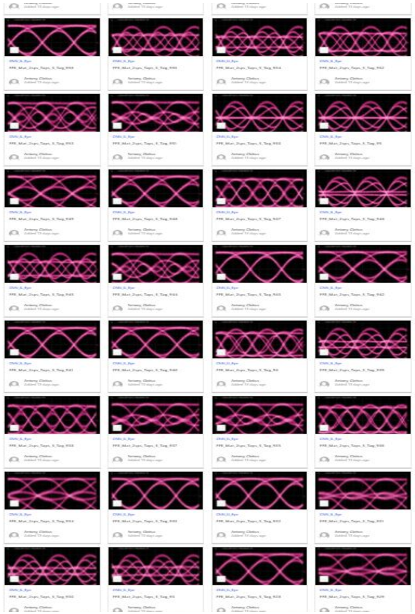


We need to develop tools for dim reduction

→ **AUTOENCODERS**

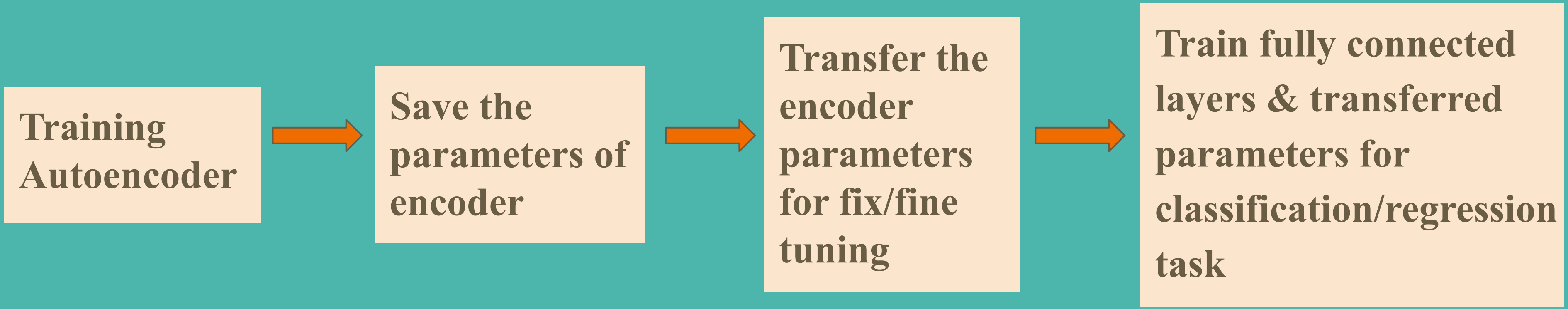
CHALLENGES

II. SMALL IMAGE DATASET AVAILABILITY



→ **TRANSFER LEARNING**

STEPS OF ACTION



Training Autoencoder

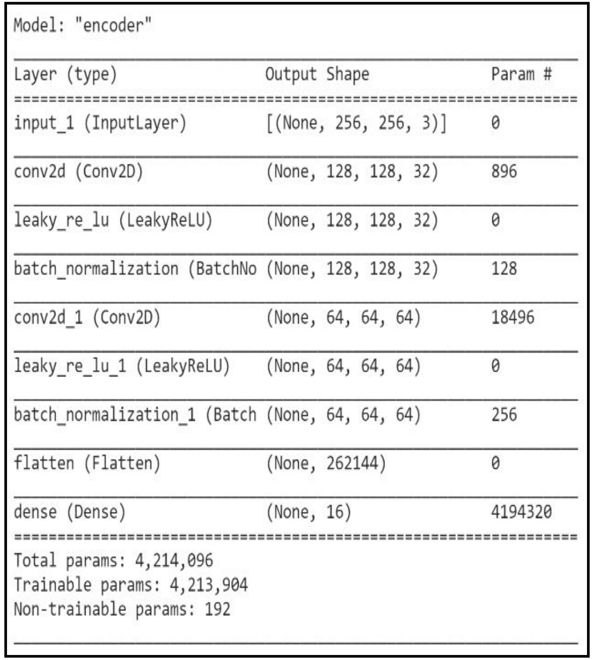
Save the parameters of encoder

Transfer the encoder parameters for fix/fine tuning

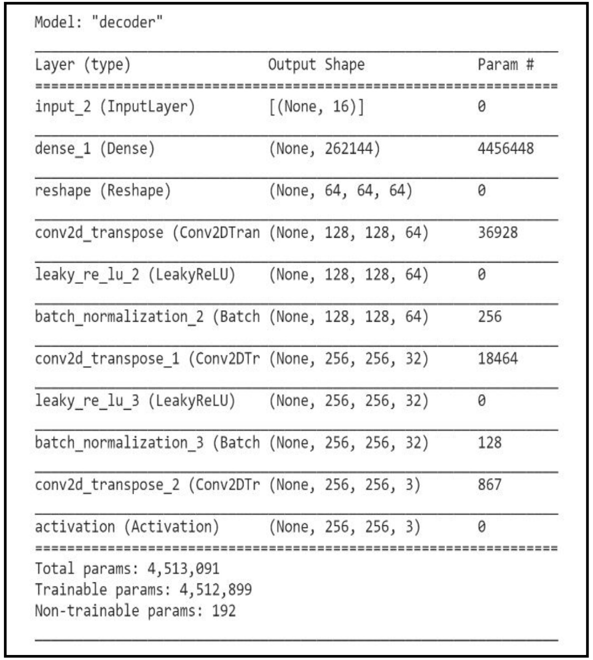
Train fully connected layers & transferred parameters for classification/regression task

MODEL- Autoencoder

Dataset Size: 1000 **Train/Val Split: 900/100** **Input Size: (256,256)**



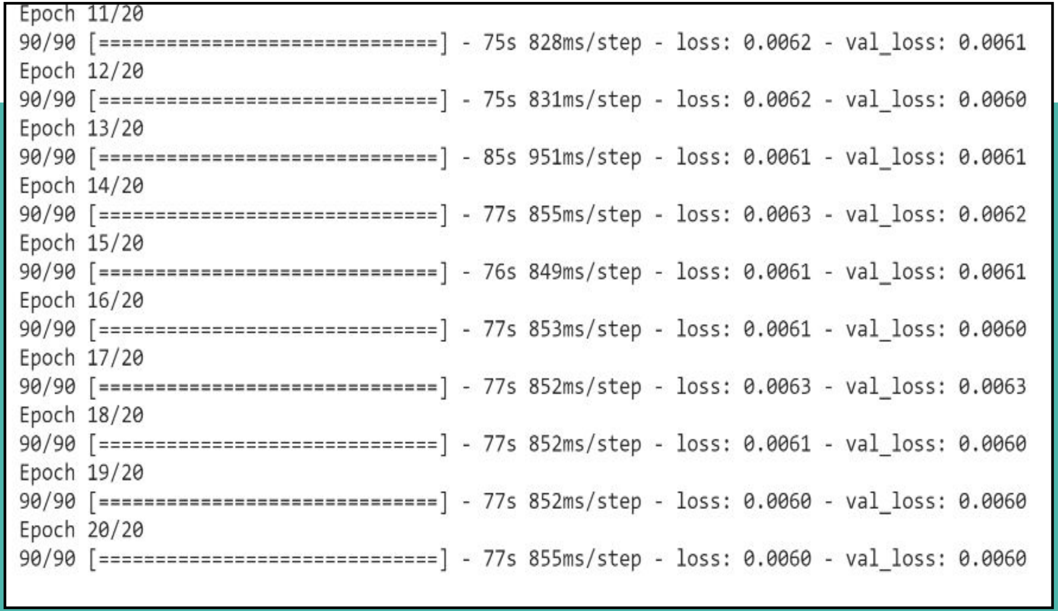
```
Model: "encoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
conv2d (Conv2D)              (None, 128, 128, 32)      896
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 128, 128, 32)      0
_________________________________________________________________
batch_normalization (BatchNo (None, 128, 128, 32)      128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 64)        18496
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64, 64, 64)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 64, 64)        256
_________________________________________________________________
flatten (Flatten)            (None, 262144)            0
_________________________________________________________________
dense (Dense)                (None, 16)                4194320
=================================================================
Total params: 4,214,096
Trainable params: 4,213,904
Non-trainable params: 192
_________________________________________________________________
```



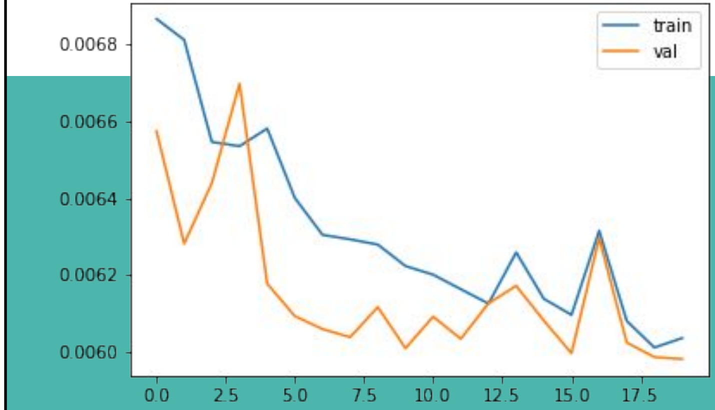
```
Model: "decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 16)]              0
_________________________________________________________________
dense_1 (Dense)              (None, 262144)            4456448
_________________________________________________________________
reshape (Reshape)            (None, 64, 64, 64)        0
_________________________________________________________________
conv2d_transpose (Conv2DTran (None, 128, 128, 64)      36928
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 128, 128, 64)      0
_________________________________________________________________
batch_normalization_2 (Batch (None, 128, 128, 64)      256
_________________________________________________________________
conv2d_transpose_1 (Conv2DTr (None, 256, 256, 32)      18464
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 256, 256, 32)      0
_________________________________________________________________
batch_normalization_3 (Batch (None, 256, 256, 32)      128
_________________________________________________________________
conv2d_transpose_2 (Conv2DTr (None, 256, 256, 3)       867
_________________________________________________________________
activation (Activation)      (None, 256, 256, 3)       0
=================================================================
Total params: 4,513,091
Trainable params: 4,512,899
Non-trainable params: 192
_________________________________________________________________
```

TRAINING- Autoencoder

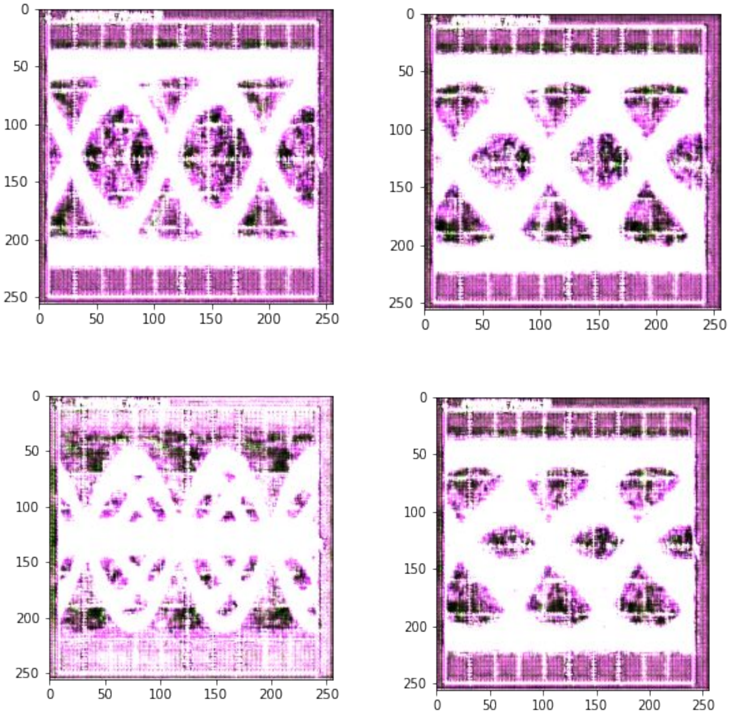
Epochs: 40 **Batch_size= 10** **Optimizer: Adam** **Loss : mse**



```
Epoch 11/20
90/90 [==============================] - 75s 828ms/step - loss: 0.0062 - val_loss: 0.0061
Epoch 12/20
90/90 [==============================] - 75s 831ms/step - loss: 0.0062 - val_loss: 0.0060
Epoch 13/20
90/90 [==============================] - 85s 951ms/step - loss: 0.0061 - val_loss: 0.0061
Epoch 14/20
90/90 [==============================] - 77s 855ms/step - loss: 0.0063 - val_loss: 0.0062
Epoch 15/20
90/90 [==============================] - 76s 849ms/step - loss: 0.0061 - val_loss: 0.0061
Epoch 16/20
90/90 [==============================] - 77s 853ms/step - loss: 0.0061 - val_loss: 0.0060
Epoch 17/20
90/90 [==============================] - 77s 852ms/step - loss: 0.0063 - val_loss: 0.0063
Epoch 18/20
90/90 [==============================] - 77s 852ms/step - loss: 0.0061 - val_loss: 0.0060
Epoch 19/20
90/90 [==============================] - 77s 852ms/step - loss: 0.0060 - val_loss: 0.0060
Epoch 20/20
90/90 [==============================] - 77s 855ms/step - loss: 0.0060 - val_loss: 0.0060
```



RESULT: Reconstructed Images

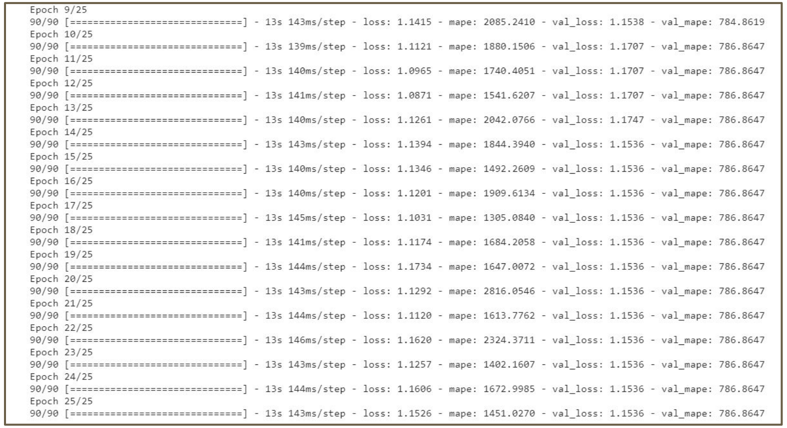


0.006

VALIDATION LOSS

REGRESSION- With Trained Encoder

Epochs: 40 **Batch_size= 10** **Optimizer: Adam** **Loss : mse**

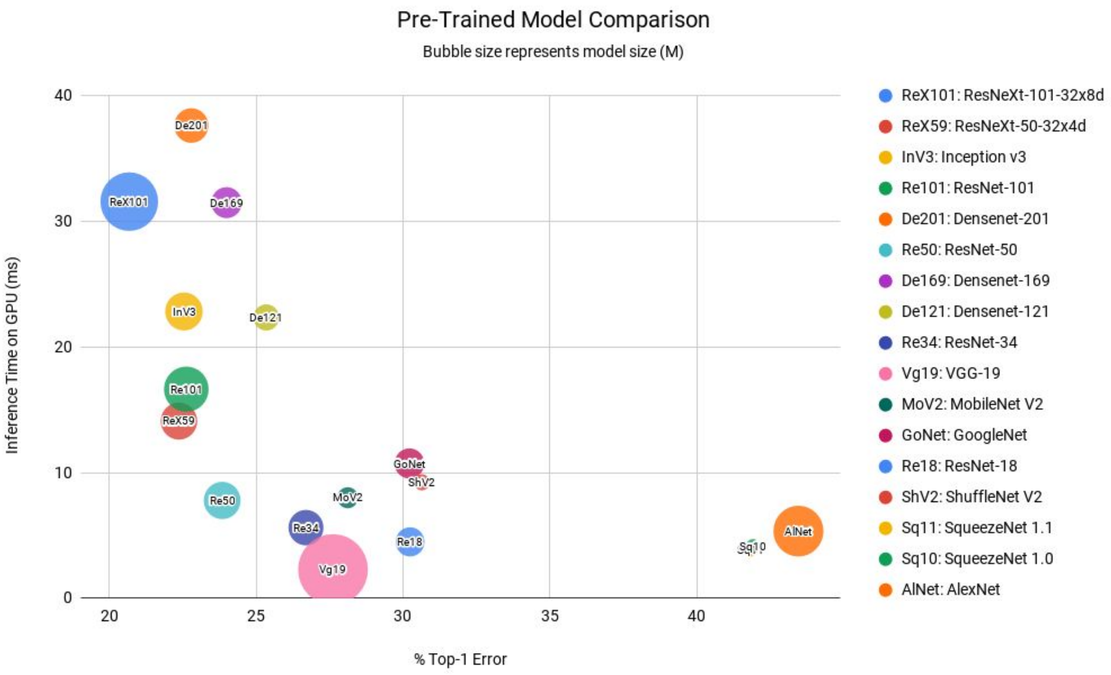


```
Epoch 9/25
90/90 [==============================] - 13s 143ms/step - loss: 1.1415 - mape: 2085.2410 - val_loss: 1.1538 - val_mape: 784.8619
Epoch 10/25
90/90 [==============================] - 13s 139ms/step - loss: 1.1121 - mape: 1880.1506 - val_loss: 1.1707 - val_mape: 786.8647
Epoch 11/25
90/90 [==============================] - 13s 140ms/step - loss: 1.0965 - mape: 1740.4051 - val_loss: 1.1707 - val_mape: 786.8647
Epoch 12/25
90/90 [==============================] - 13s 141ms/step - loss: 1.0871 - mape: 1541.6207 - val_loss: 1.1707 - val_mape: 786.8647
Epoch 13/25
90/90 [==============================] - 13s 140ms/step - loss: 1.1261 - mape: 2042.0766 - val_loss: 1.1747 - val_mape: 786.8647
Epoch 14/25
90/90 [==============================] - 13s 143ms/step - loss: 1.1394 - mape: 1844.3940 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 15/25
90/90 [==============================] - 13s 140ms/step - loss: 1.1346 - mape: 1492.2609 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 16/25
90/90 [==============================] - 13s 140ms/step - loss: 1.1201 - mape: 1909.6134 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 17/25
90/90 [==============================] - 13s 145ms/step - loss: 1.1031 - mape: 1305.0840 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 18/25
90/90 [==============================] - 13s 141ms/step - loss: 1.1174 - mape: 1684.2058 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 19/25
90/90 [==============================] - 13s 144ms/step - loss: 1.1734 - mape: 1647.0072 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 20/25
90/90 [==============================] - 13s 143ms/step - loss: 1.1292 - mape: 2816.0546 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 21/25
90/90 [==============================] - 13s 144ms/step - loss: 1.1120 - mape: 1613.7762 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 22/25
90/90 [==============================] - 13s 146ms/step - loss: 1.1620 - mape: 2324.3711 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 23/25
90/90 [==============================] - 13s 143ms/step - loss: 1.1257 - mape: 1402.1607 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 24/25
90/90 [==============================] - 13s 144ms/step - loss: 1.1606 - mape: 1672.9985 - val_loss: 1.1536 - val_mape: 786.8647
Epoch 25/25
90/90 [==============================] - 13s 143ms/step - loss: 1.1526 - mape: 1451.0270 - val_loss: 1.1536 - val_mape: 786.8647
```

1.15

VALIDATION LOSS

→ **MODEL OVERFITTING**



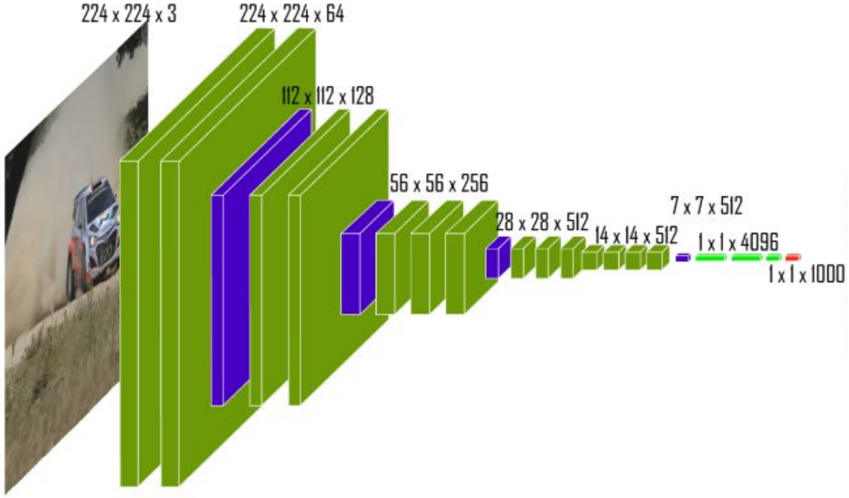
Pre-Trained Model Comparison

Bubble size represents model size (M)

Regression with previously trained models

MODELS AND ACCURACIES:

I. VGG16



INPUT SIZE: (150,150)

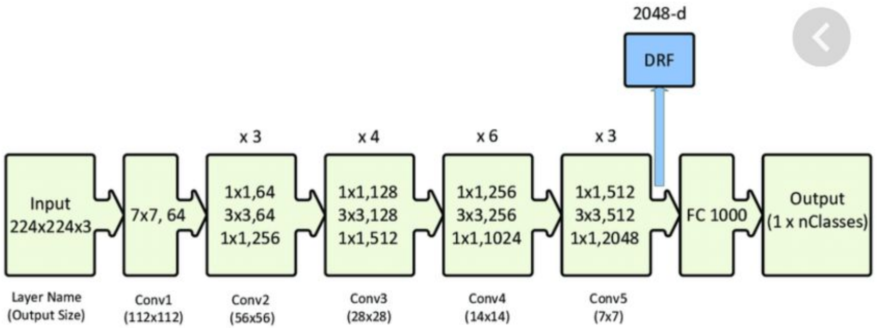
EPOCHS: 30

TRAIN_LOSS: 0.0832

VAL_LOSS: 0.1491

MODELS AND ACCURACIES:

II. RESNET50



INPUT SIZE: (150,150)

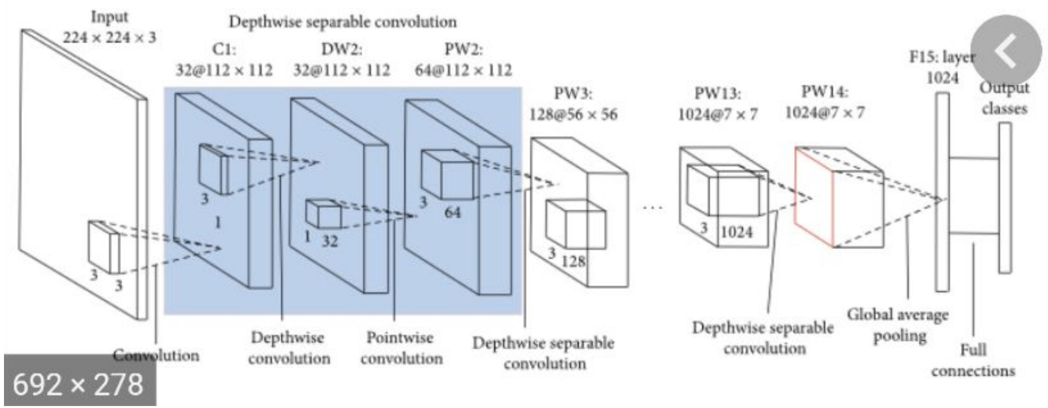
EPOCHS: 30

TRAIN_LOSS: 0.1580

VAL_LOSS: 0.1495

MODELS AND ACCURACIES:

III. MOBILENET



INPUT SIZE: (150,150)

EPOCHS: 200

TRAIN_LOSS: 0.0103

VAL_LOSS: 0.1494

What’s Next?

Plan for the next few weeks

- Train autoencoder on a larger, different dataset & TL its parameters
- Increase the eye dataset
- Try other models (Like DenseNets, Inception etc.)
- Explore better performing models for regression

Learning Experience

- Tackling the problems related to small datasets
- Handling regression tasks in Computer Vision
- Exploring ways to avoid overfitting
- Discovering signal characteristics and distortions by eye diagram inspection

Thank You