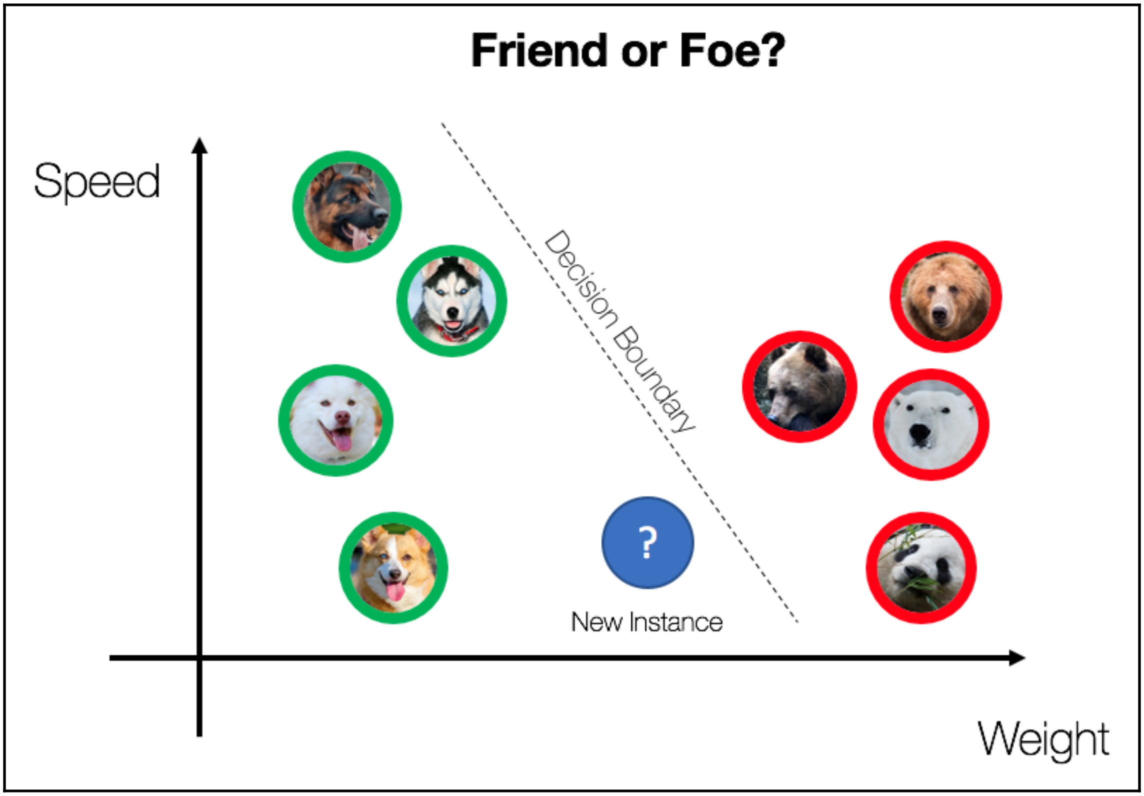
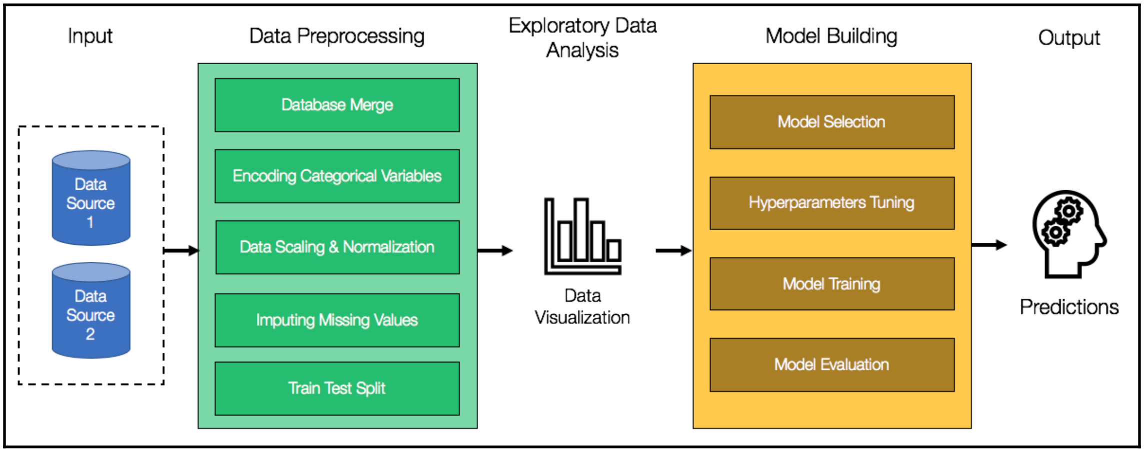
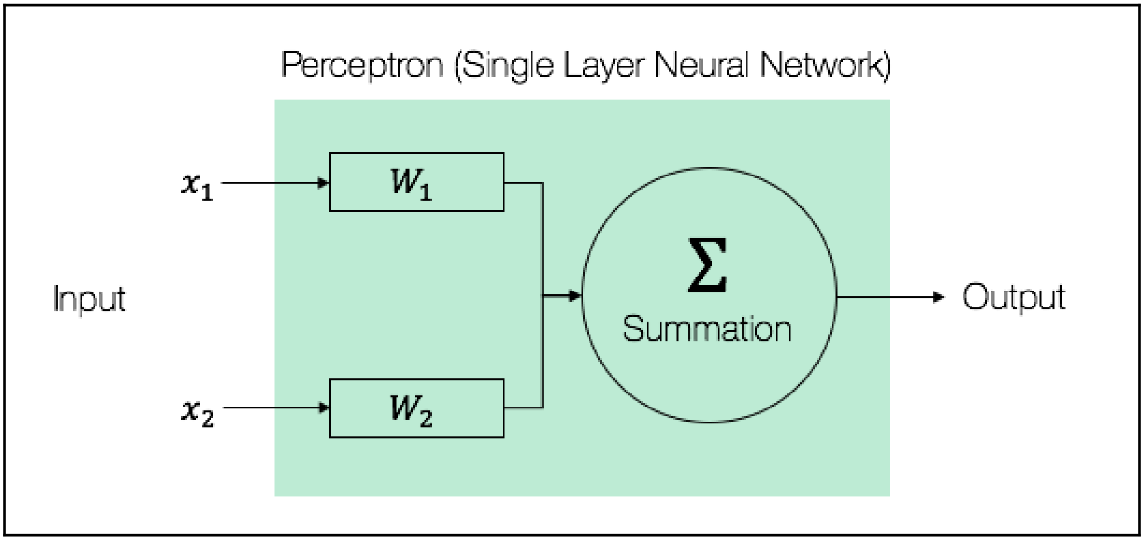
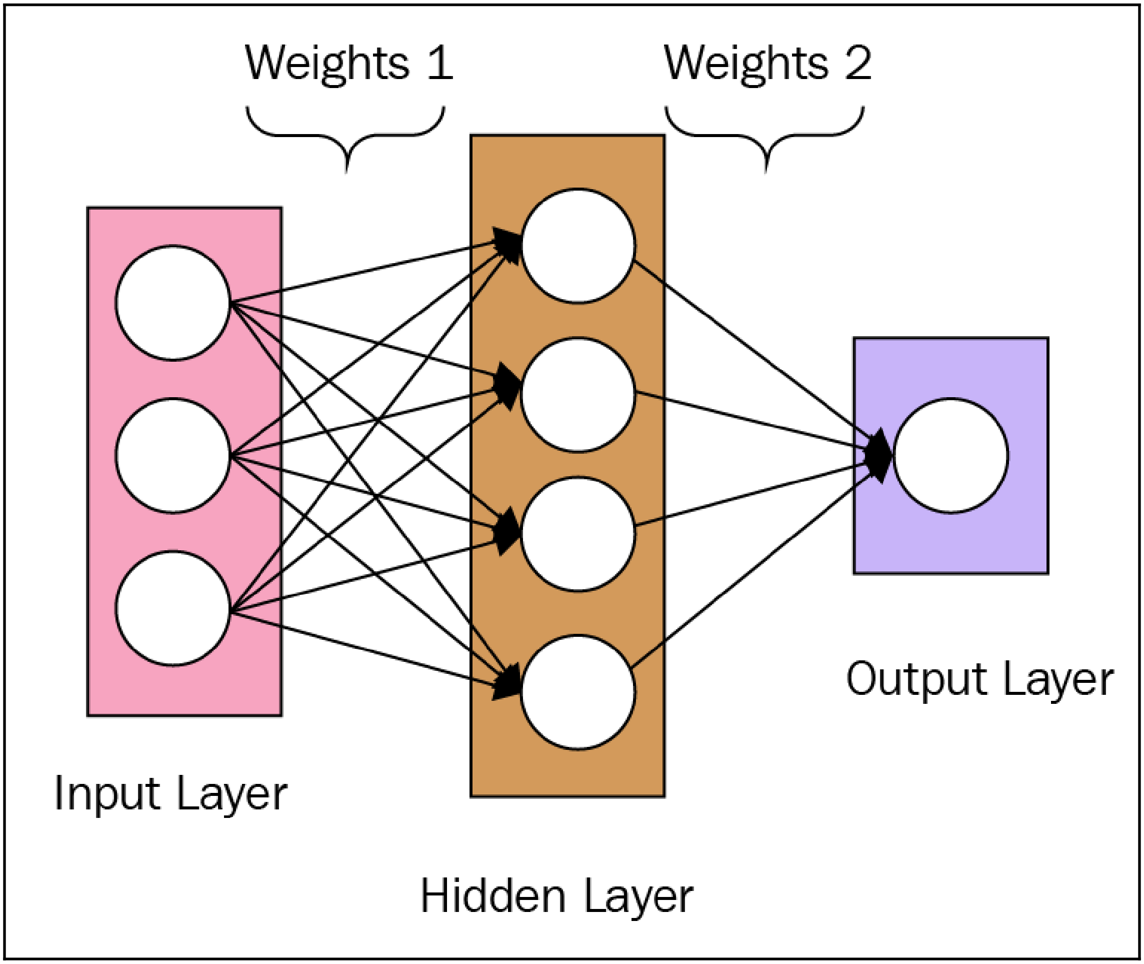
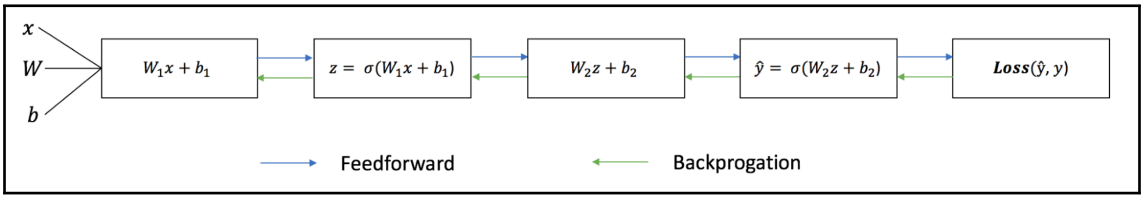
Chapter 1: Machine Learning and Neural Networks 101

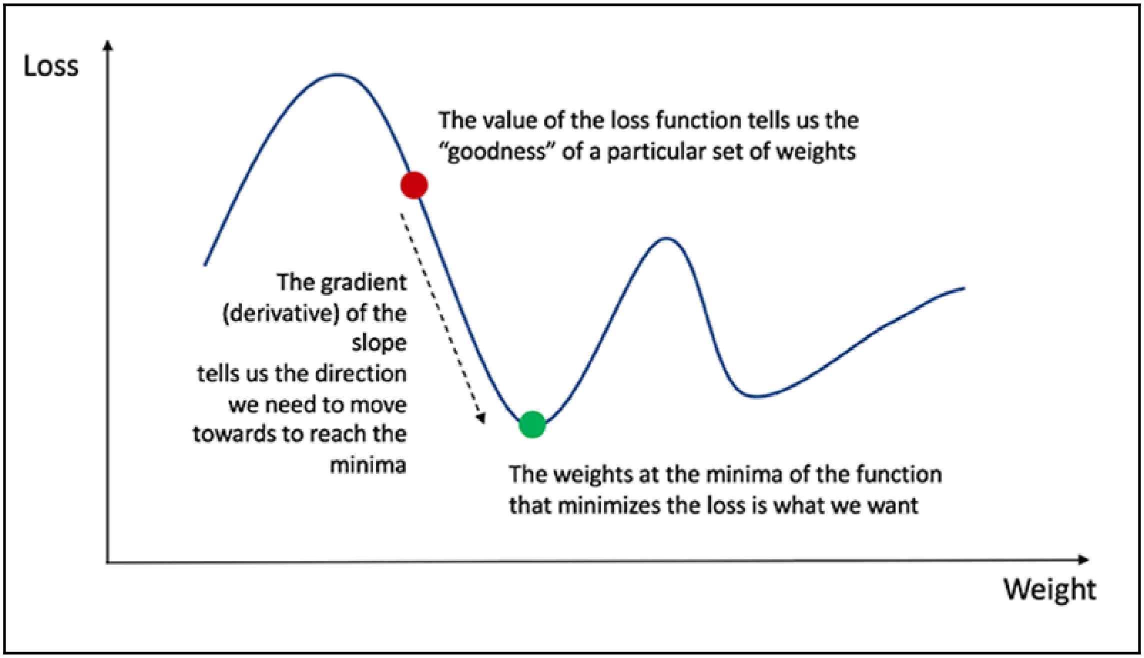



Input
Data Preprocessing
Exploratory Data Analysis
Model Building
Output
Data Source 1
Data Source 2
Database Merge
Encoding Categorical Variables
Data Scaling & Normalization
Imputing Missing Values
Train Test Split
Data Visualization
Model Selection
Hyperparameters Tuning
Model Training
Model Evaluation
Predictions


Perceptron (Single Layer Neural Network)
Input
x_1
x_2
W_1
W_2
Σ
Summation
Output


Weights 1
Weights 2
Input Layer
Hidden Layer
Output Layer


x
W
b
$W_1x + b_1$
$z = \sigma(W_1x + b_1)$
$W_2z + b_2$
$\hat{y} = \sigma(W_2z + b_2)$
$\boldsymbol{Loss}(\hat{y}, y)$
Feedforward
Backprogation


Loss
The value of the loss function tells us the
"goodness" of a particular set of weights
The gradient
(derivative) of the
slope
tells us the direction
we need to move
towards to reach the
minima
The weights at the minima of the function
that minimizes the loss is what we want
Weight

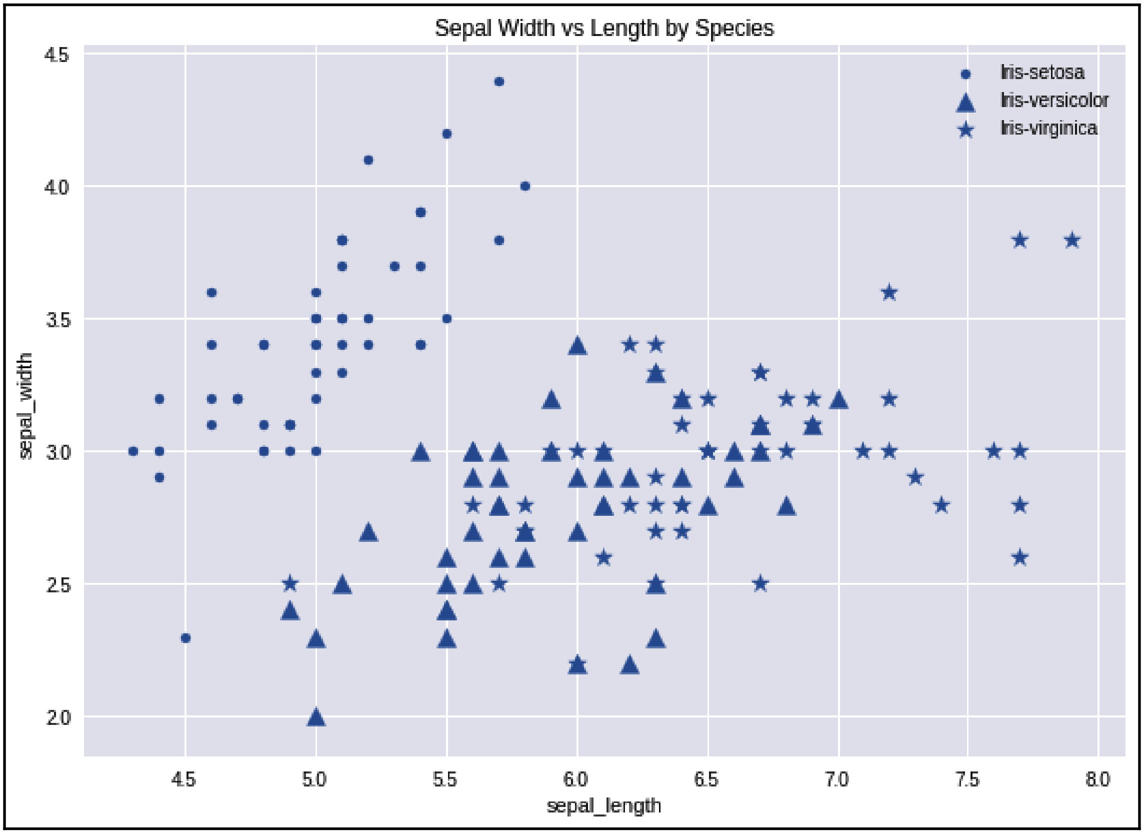

Loss per Iteration
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
Loss
0
100
200
300
400
500
600
700
800
900
1000
1100
1200
1300
1400
1500
Iteration

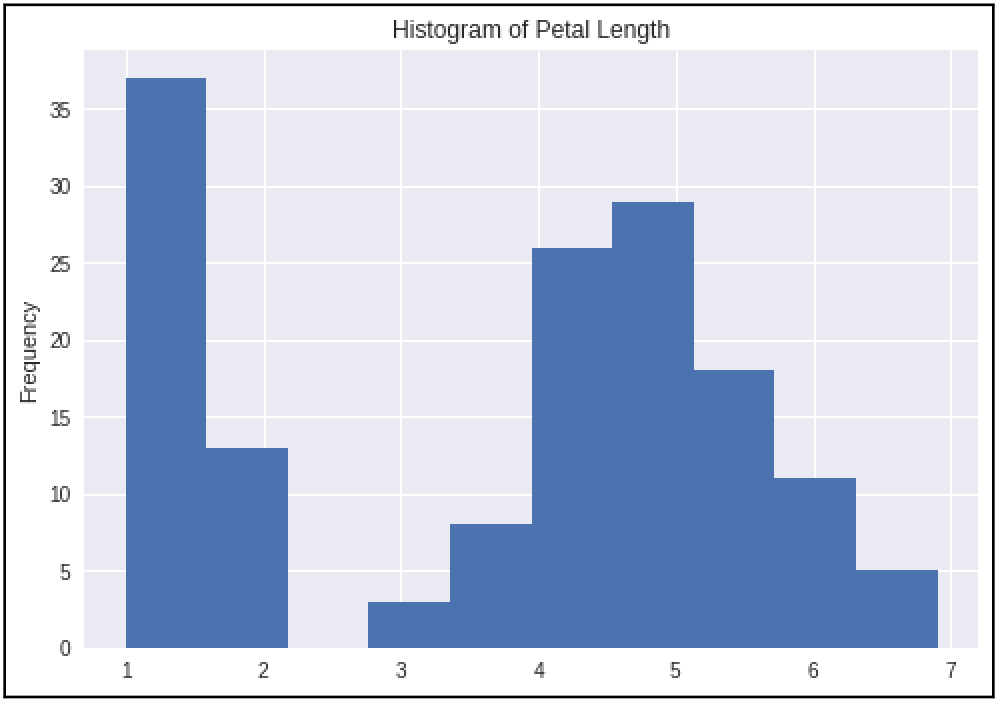
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
sepal_length    150 non-null float64
sepal_width     150 non-null float64
petal_length    150 non-null float64
petal_width     150 non-null float64
class           150 non-null object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
6	4.6	3.4	1.4	0.3	Iris-setosa
7	5.0	3.4	1.5	0.2	Iris-setosa
8	4.4	2.9	1.4	0.2	Iris-setosa
9	4.9	3.1	1.5	0.1	Iris-setosa

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa
10	5.4	3.7	1.5	0.2	Iris-setosa
14	5.8	4.0	1.2	0.2	Iris-setosa
15	5.7	4.4	1.5	0.4	Iris-setosa
16	5.4	3.9	1.3	0.4	Iris-setosa
17	5.1	3.5	1.4	0.3	Iris-setosa
18	5.7	3.8	1.7	0.3	Iris-setosa
19	5.1	3.8	1.5	0.3	Iris-setosa
20	5.4	3.4	1.7	0.2	Iris-setosa


Sepal Width vs Length by Species
Iris-setosa
Iris-versicolor
Iris-virginica
sepal_width
sepal_length
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0


Histogram of Petal Length
Frequency
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7

Original Features

	Day
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

One-Hot-Encoding

Transformed Features

	Day_Monday	Day_Tuesday	Day_Wednesday	Day_Thursday	Day_Friday	Day_Saturday	Day_Sunday
1	1	0	0	0	0	0	0
2	0	1	0	0	0	0	0
3	0	0	1	0	0	0	0
4	0	0	0	1	0	0	0
5	0	0	0	0	1	0	0
6	0	0	0	0	0	1	0
7	0	0	0	0	0	0	1

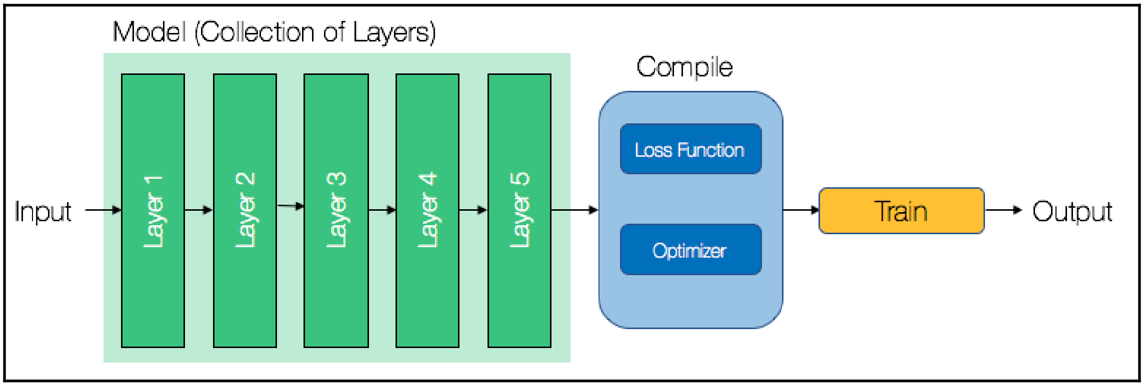
	Day
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

	Day_Friday	Day_Monday	Day_Saturday	Day_Sunday	Day_Thursday	Day_Tuesday	Day_Wednesday
0	0	1	0	0	0	0	0
1	0	0	0	0	0	1	0
2	0	0	0	0	0	0	1
3	0	0	0	0	1	0	0
4	1	0	0	0	0	0	0
5	0	0	1	0	0	0	0
6	0	0	0	1	0	0	0

```
sepal_length      True
sepal_width      False
petal_length     False
petal_width      False
class            False
dtype: bool
```

```
Number of rows before deleting: 150
Number of rows after  deleting: 140
```

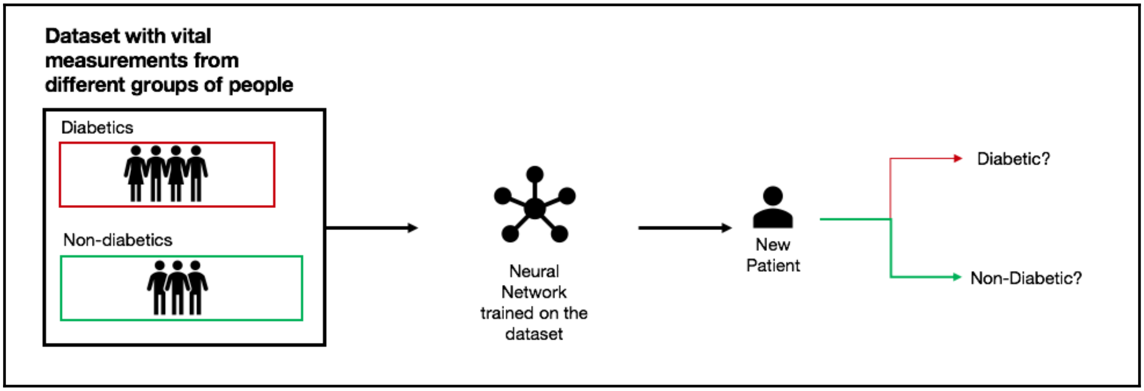
```
sepal_length      False
sepal_width       False
petal_length      False
petal_width       False
class             False
dtype: bool
```

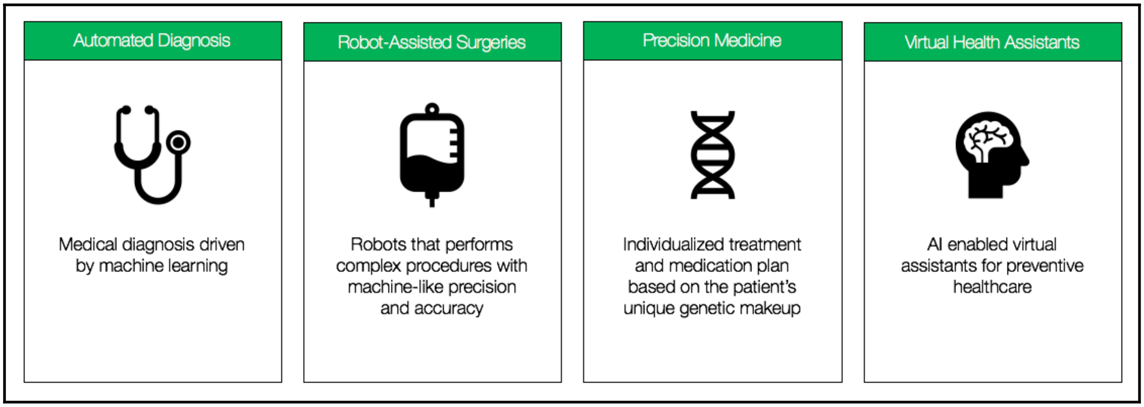



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 4)                 16
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 5
=================================================================
Total params: 21
Trainable params: 21
Non-trainable params: 0
```

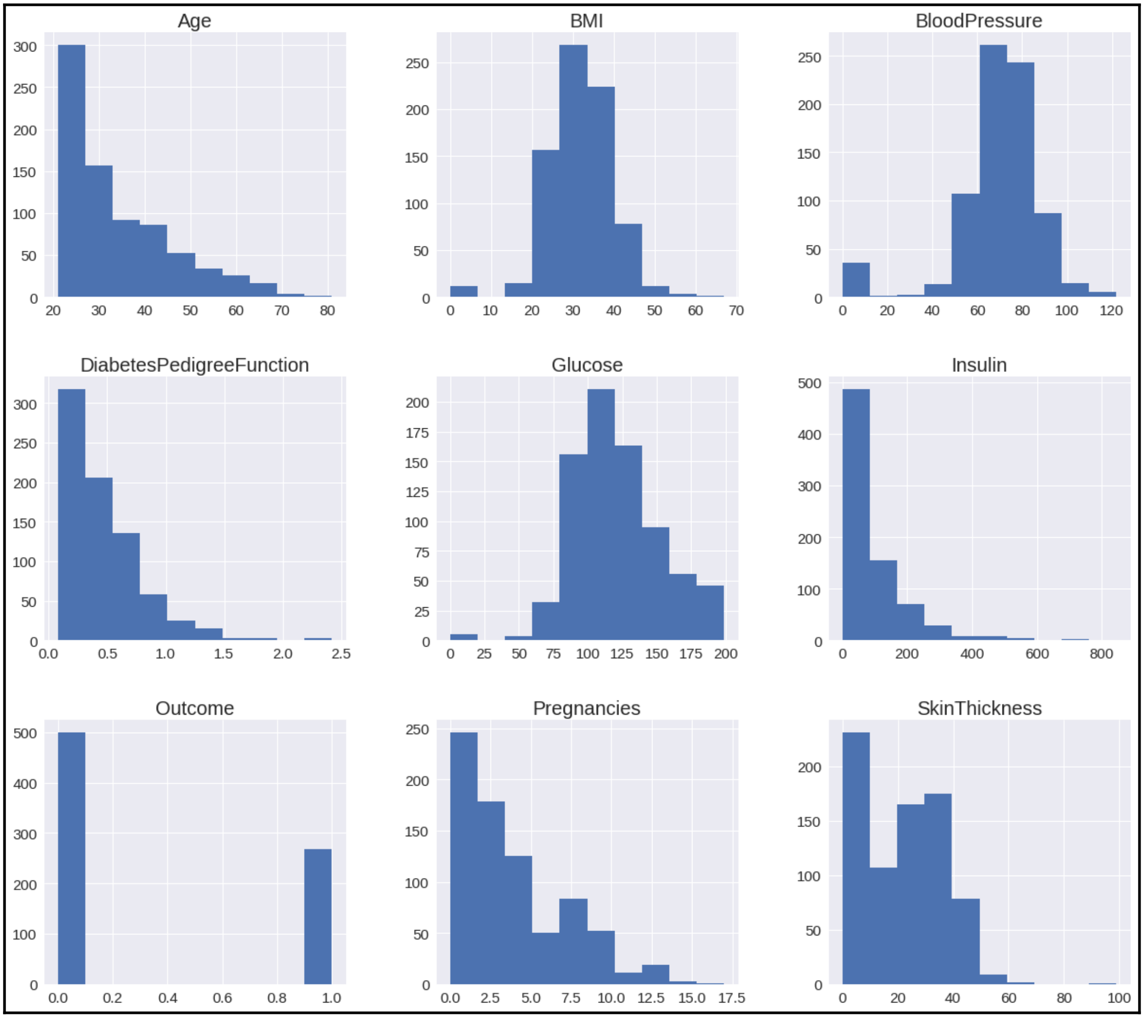
```
[[0.04623432]
 [0.94387746]
 [0.94575524]
 [0.06039287]]
```

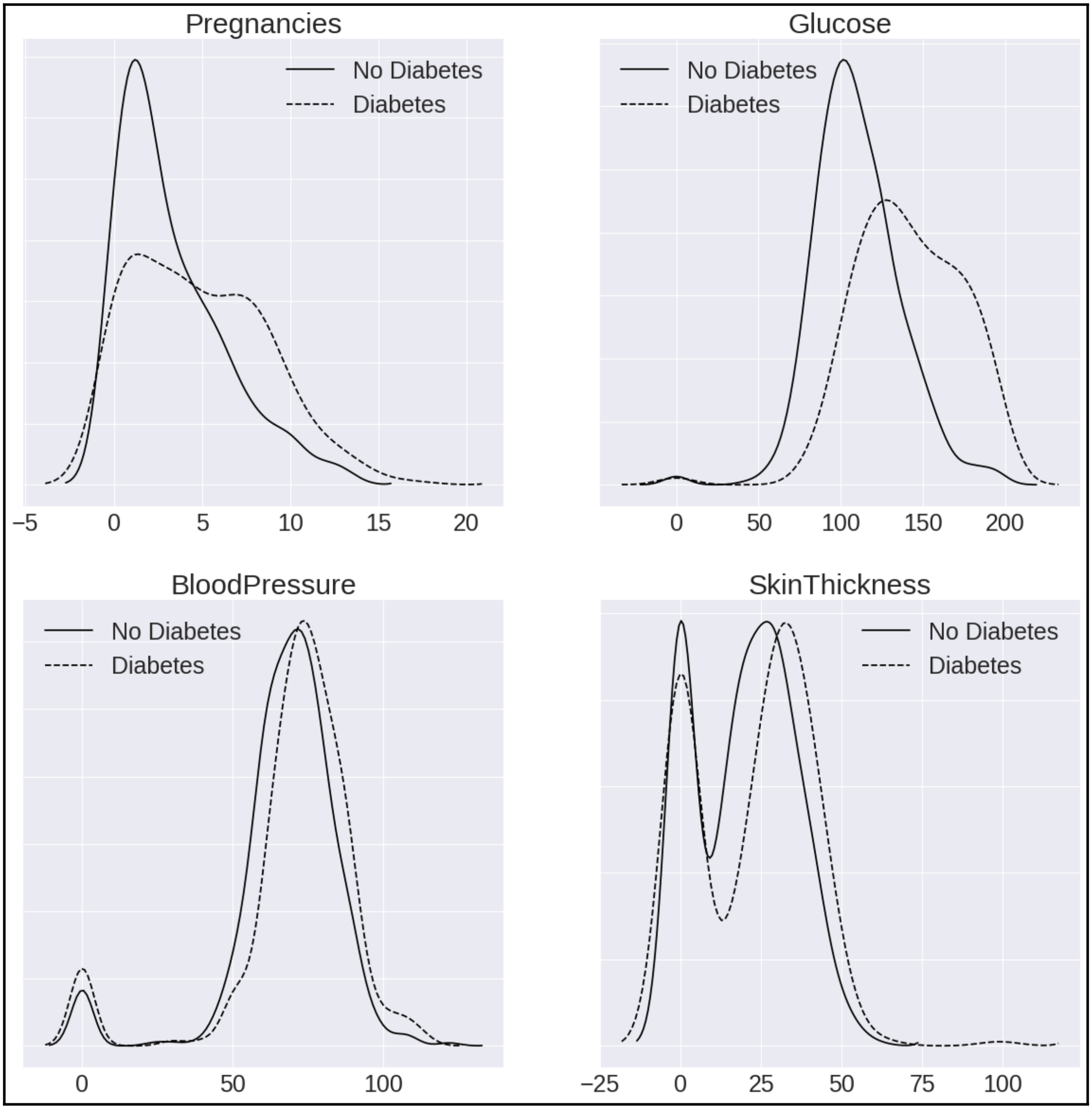
Chapter 2: Predicting Diabetes with Multilayer Perceptrons

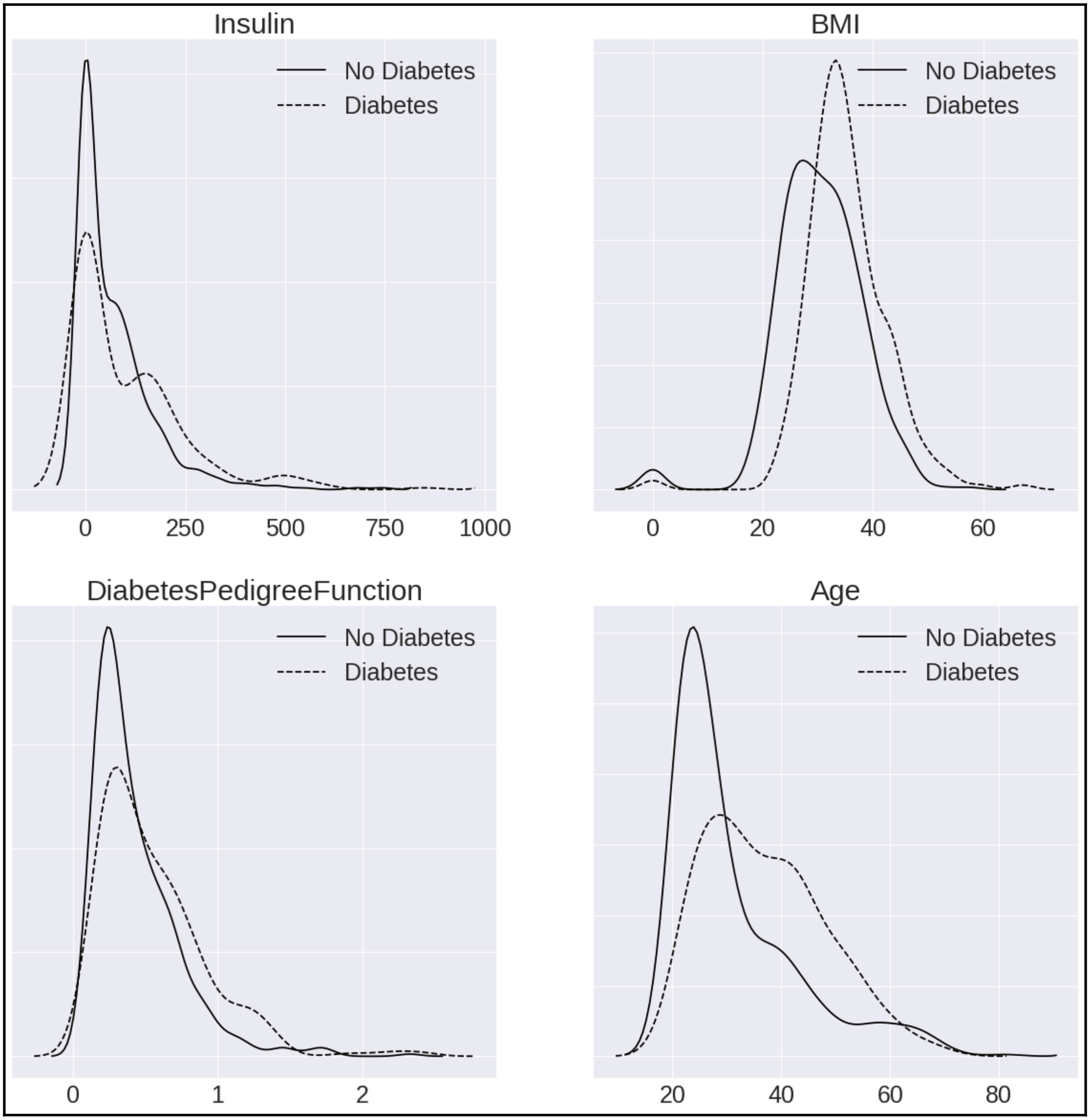




	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1


Age
BMI
BloodPressure
DiabetesPedigreeFunction
Glucose
Insulin
Outcome
Pregnancies
SkinThickness


Pregnancies
No Diabetes
Diabetes
−5
0
5
10
15
20
Glucose
No Diabetes
Diabetes
0
50
100
150
200
BloodPressure
No Diabetes
Diabetes
0
50
100
SkinThickness
No Diabetes
Diabetes
−25
0
25
50
75
100


Insulin
No Diabetes
Diabetes
0
250
500
750
1000
BMI
No Diabetes
Diabetes
0
20
40
60
DiabetesPedigreeFunction
No Diabetes
Diabetes
0
1
2
Age
No Diabetes
Diabetes
20
40
60
80

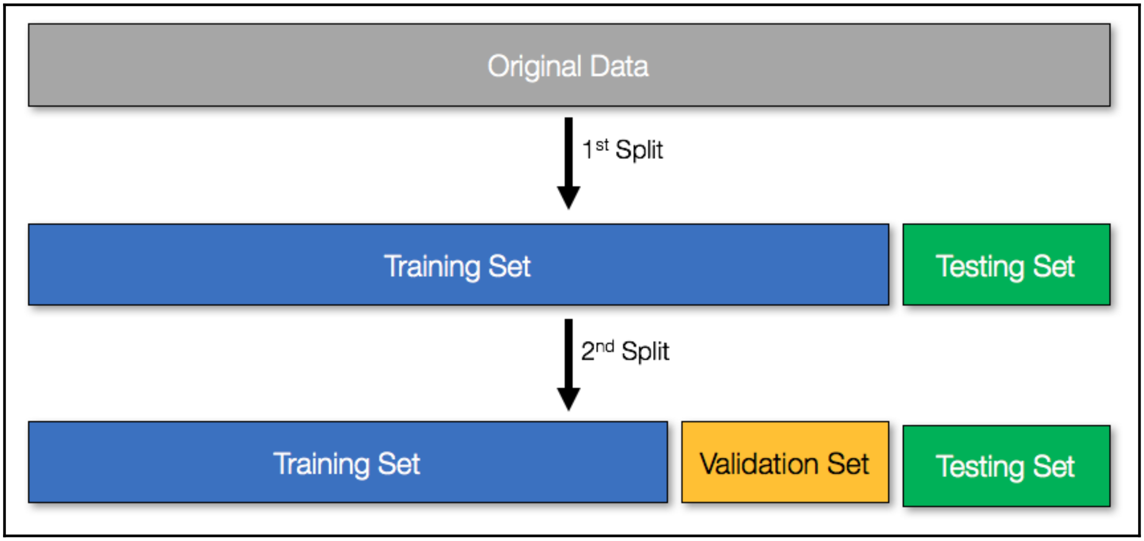
```
Pregnancies                 False
Glucose                     False
BloodPressure               False
SkinThickness               False
Insulin                     False
BMI                         False
DiabetesPedigreeFunction    False
Age                         False
Outcome                     False
dtype: bool
```

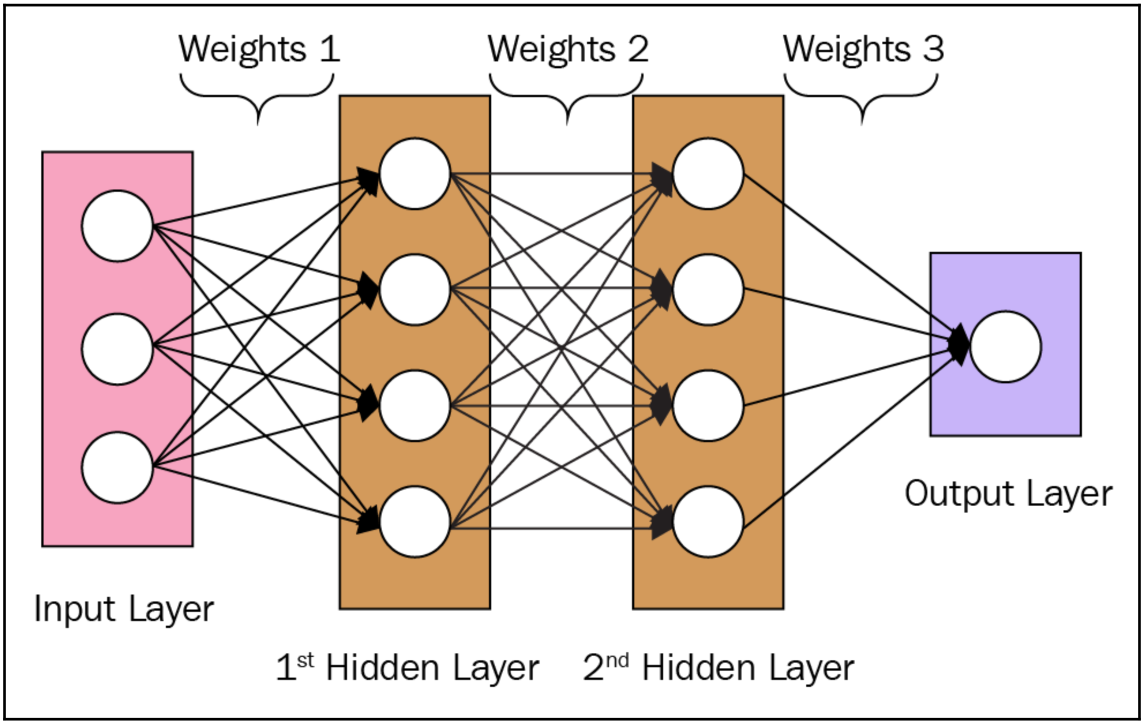
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

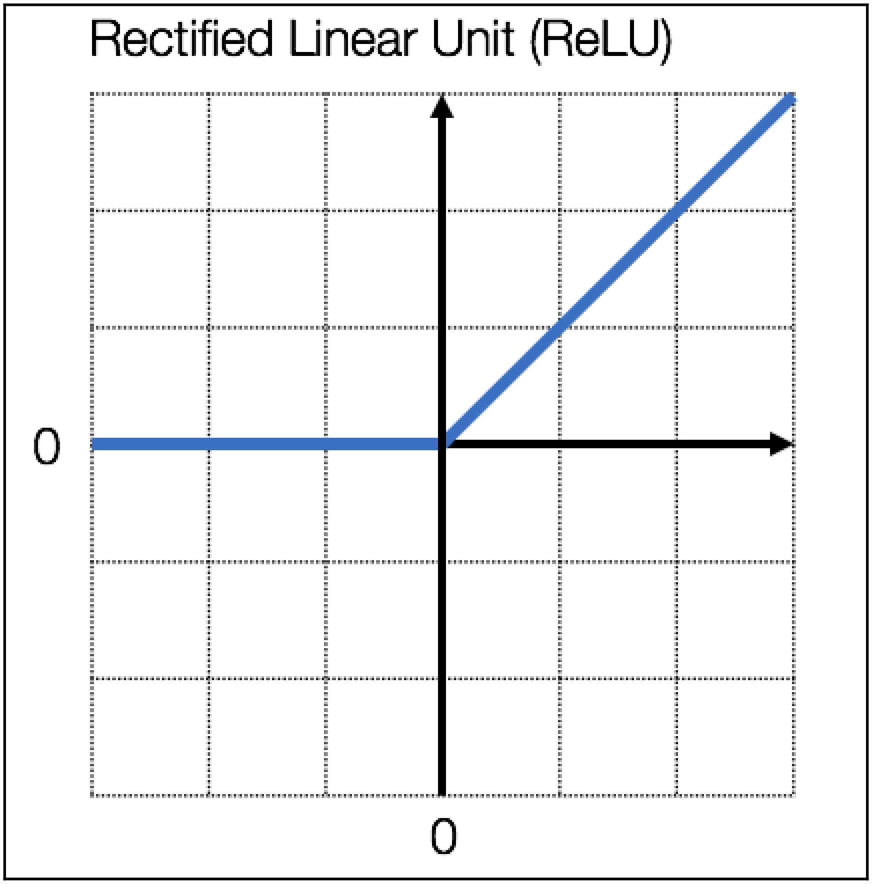
```
Number of rows with 0 values for each variable
Pregnancies: 111
Glucose: 5
BloodPressure: 35
SkinThickness: 227
Insulin: 374
BMI: 11
DiabetesPedigreeFunction: 0
Age: 0
```

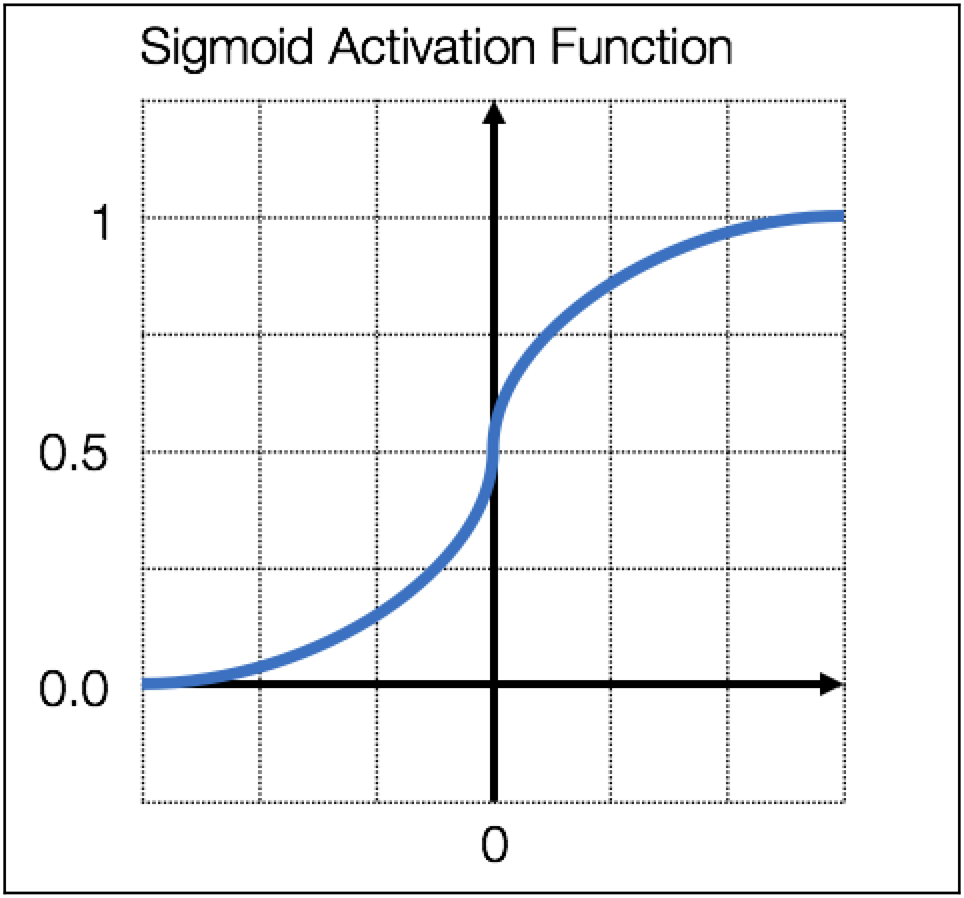
```
Number of rows with 0 values for each variable
Pregnancies: 111
Glucose: 0
BloodPressure: 0
SkinThickness: 0
Insulin: 0
BMI: 0
DiabetesPedigreeFunction: 0
Age: 0
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
mean	0.00	0.00	0.0	0.00	0.00	0.00	0.00	0.00
std	1.00	1.00	1.0	1.00	1.00	1.00	1.00	1.00
max	3.91	2.54	4.1	7.95	8.13	5.04	5.88	4.06


Original Data
1st Split
Training Set
Testing Set
2nd Split
Training Set
Validation Set
Testing Set

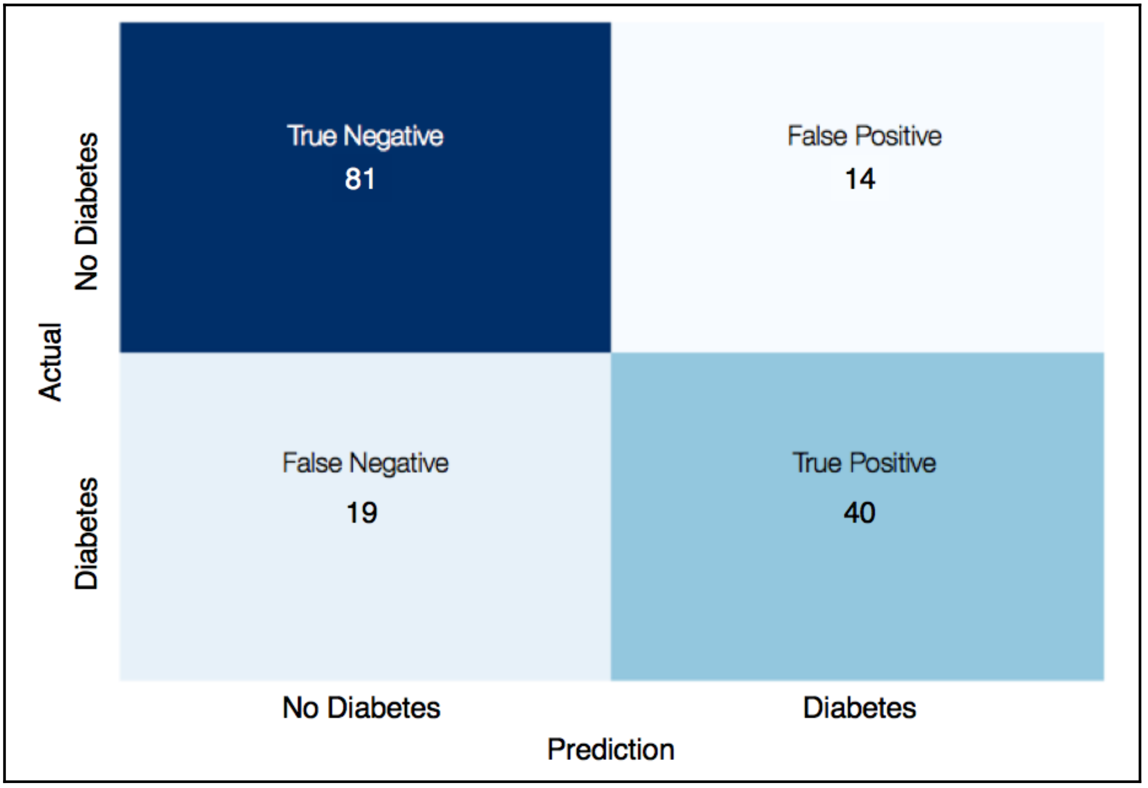

Weights 1
Weights 2
Weights 3
Input Layer
1st Hidden Layer
2nd Hidden Layer
Output Layer

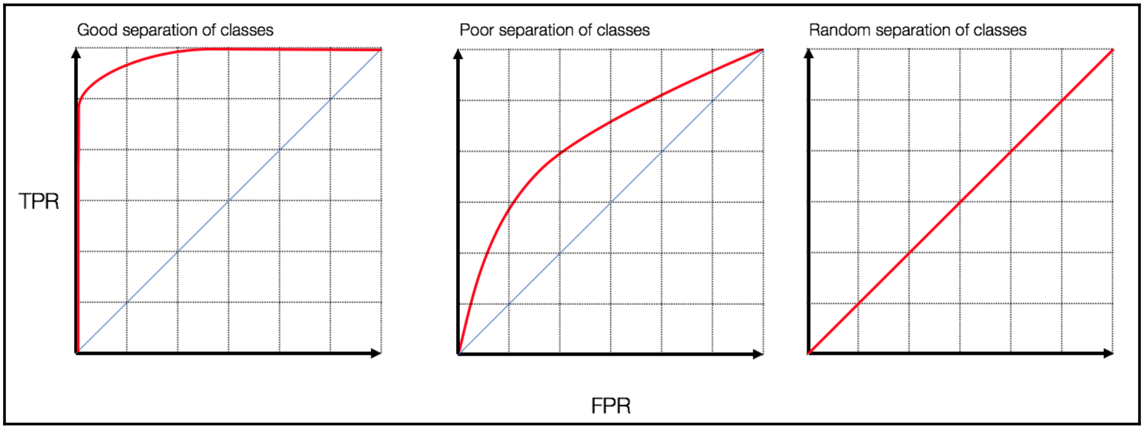

Rectified Linear Unit (ReLU)
0
0

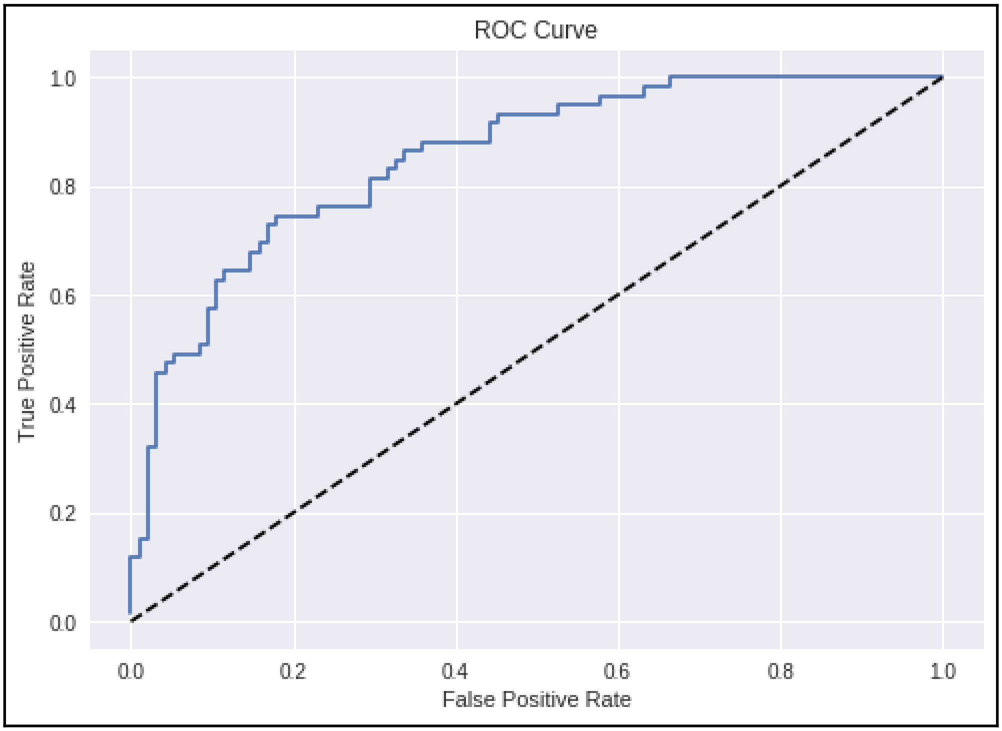

Sigmoid Activation Function
1
0.5
0.0
0

```
Epoch 1/200
491/491 [==============================] - 1s 1ms/step - loss: 0.6387 - acc: 0.6640
Epoch 2/200
491/491 [==============================] - 0s 30us/step - loss: 0.5772 - acc: 0.7189
Epoch 3/200
491/491 [==============================] - 0s 32us/step - loss: 0.5410 - acc: 0.7332
Epoch 4/200
491/491 [==============================] - 0s 35us/step - loss: 0.5159 - acc: 0.7434
Epoch 5/200
491/491 [==============================] - 0s 32us/step - loss: 0.4976 - acc: 0.7617
Epoch 6/200
491/491 [==============================] - 0s 29us/step - loss: 0.4869 - acc: 0.7597
Epoch 7/200
491/491 [==============================] - 0s 32us/step - loss: 0.4770 - acc: 0.7617
Epoch 8/200
491/491 [==============================] - 0s 32us/step - loss: 0.4697 - acc: 0.7637
Epoch 9/200
491/491 [==============================] - 0s 32us/step - loss: 0.4642 - acc: 0.7678
Epoch 10/200
491/491 [==============================] - 0s 31us/step - loss: 0.4600 - acc: 0.7658

                                      .

                                      .

                                      .

Epoch 190/200
491/491 [==============================] - 0s 31us/step - loss: 0.2488 - acc: 0.8941
Epoch 191/200
491/491 [==============================] - 0s 30us/step - loss: 0.2476 - acc: 0.9002
Epoch 192/200
491/491 [==============================] - 0s 30us/step - loss: 0.2492 - acc: 0.8982
Epoch 193/200
491/491 [==============================] - 0s 30us/step - loss: 0.2466 - acc: 0.9022
Epoch 194/200
491/491 [==============================] - 0s 33us/step - loss: 0.2476 - acc: 0.8961
Epoch 195/200
491/491 [==============================] - 0s 32us/step - loss: 0.2490 - acc: 0.8921
Epoch 196/200
491/491 [==============================] - 0s 36us/step - loss: 0.2473 - acc: 0.8961
Epoch 197/200
491/491 [==============================] - 0s 32us/step - loss: 0.2455 - acc: 0.8961
Epoch 198/200
491/491 [==============================] - 0s 35us/step - loss: 0.2422 - acc: 0.9022
Epoch 199/200
491/491 [==============================] - 0s 31us/step - loss: 0.2428 - acc: 0.8961
Epoch 200/200
491/491 [==============================] - 0s 32us/step - loss: 0.2412 - acc: 0.9022
```

```
491/491 [==============================] - 0s 45us/step
Training Accuracy: 91.85%

154/154 [==============================] - 0s 48us/step
Testing Accuracy: 78.57%
```

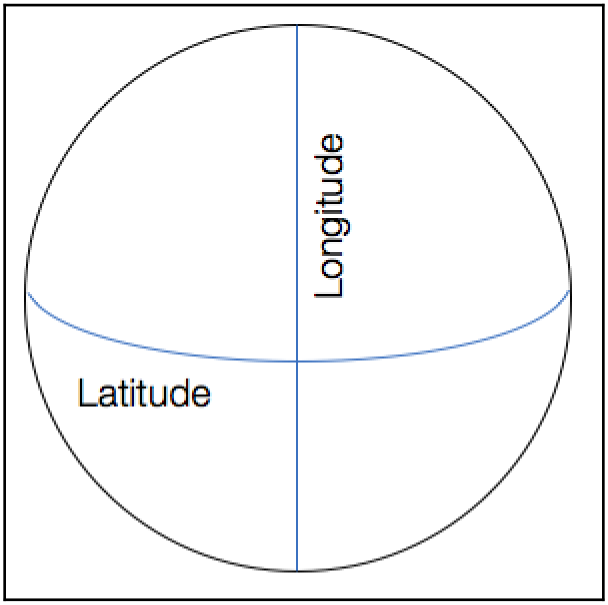


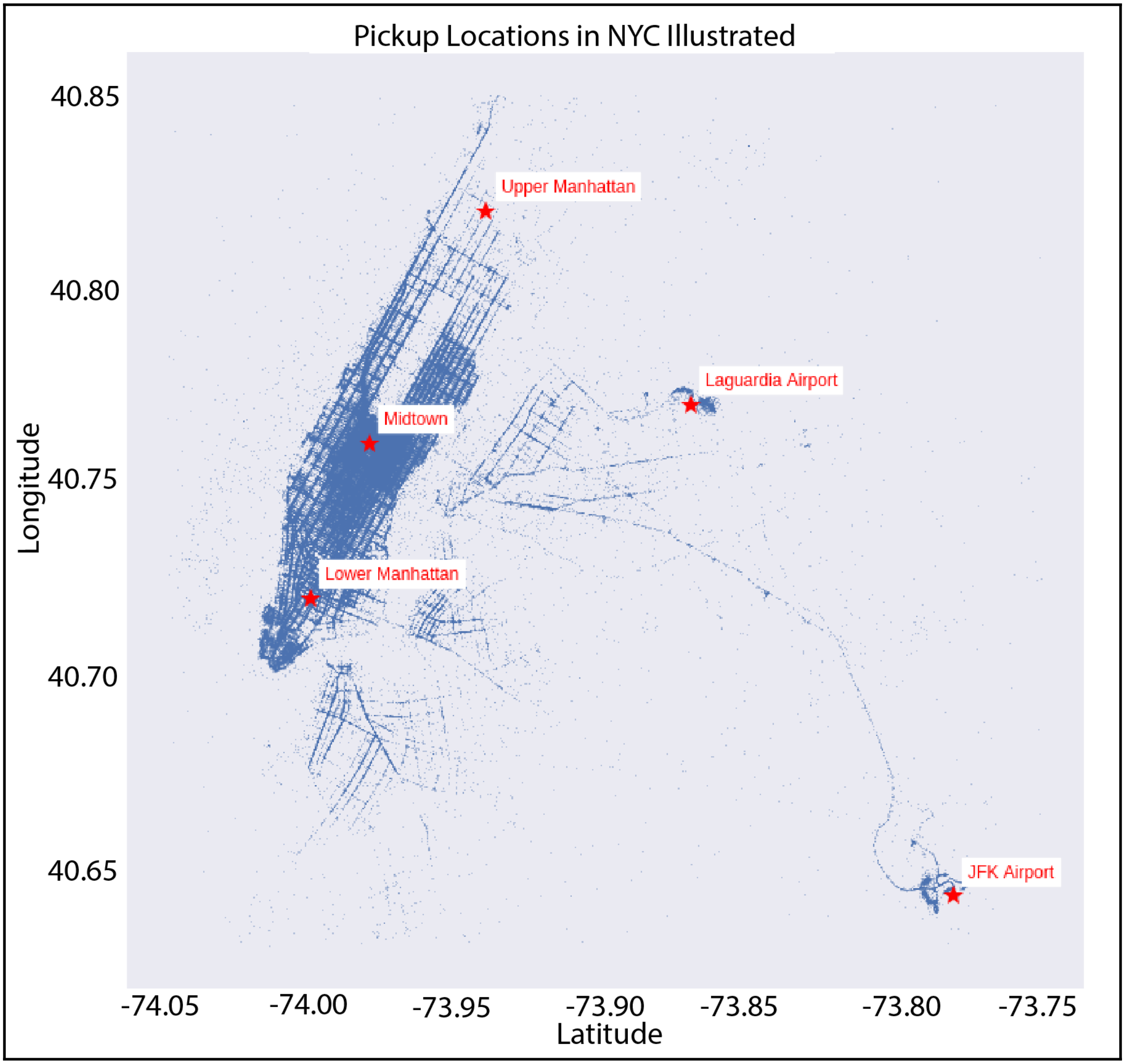

Good separation of classes
Poor separation of classes
Random separation of classes
TPR
FPR


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

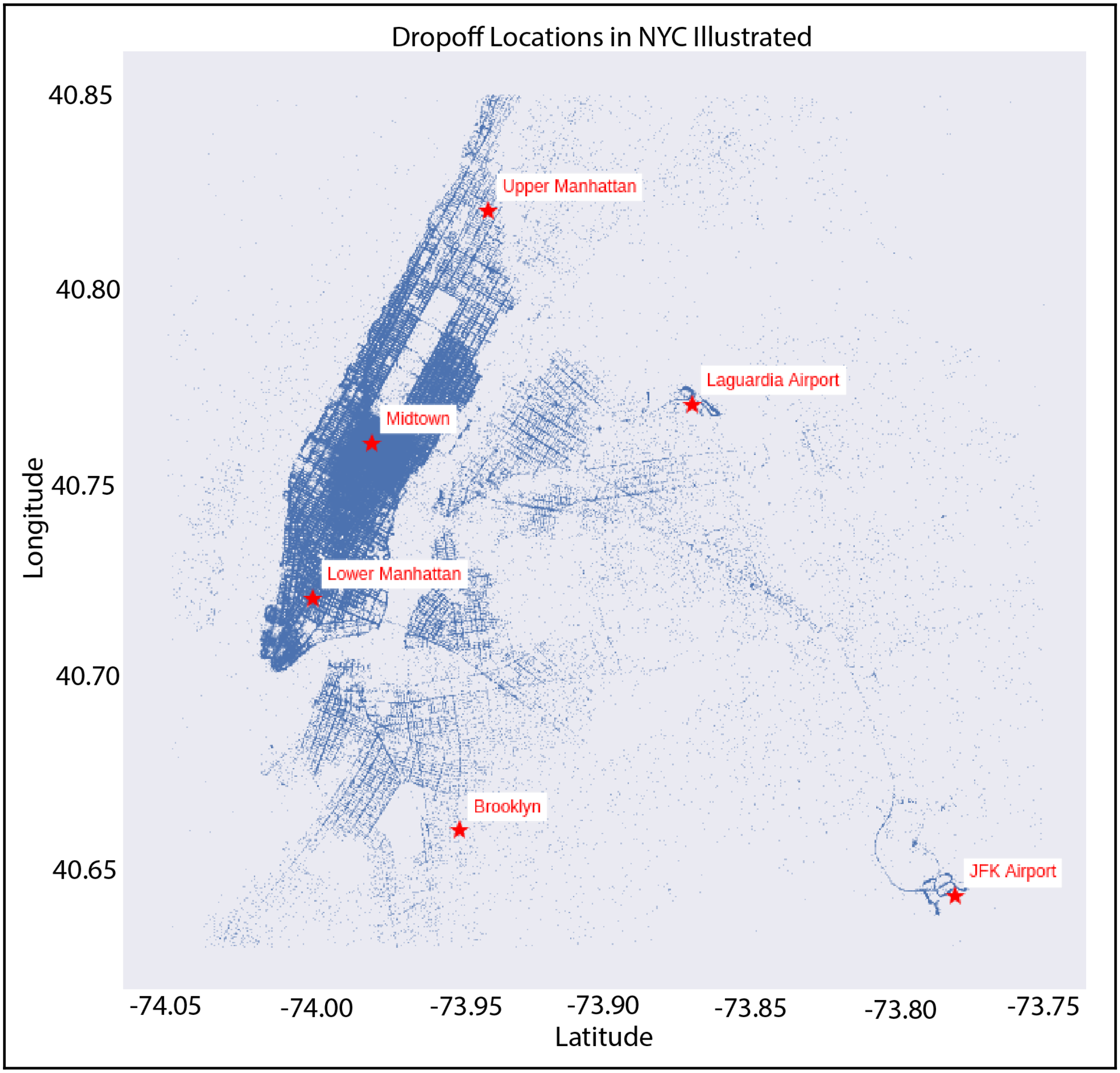
Chapter 3: Predicting Taxi Fares with Deep Feedforward Networks

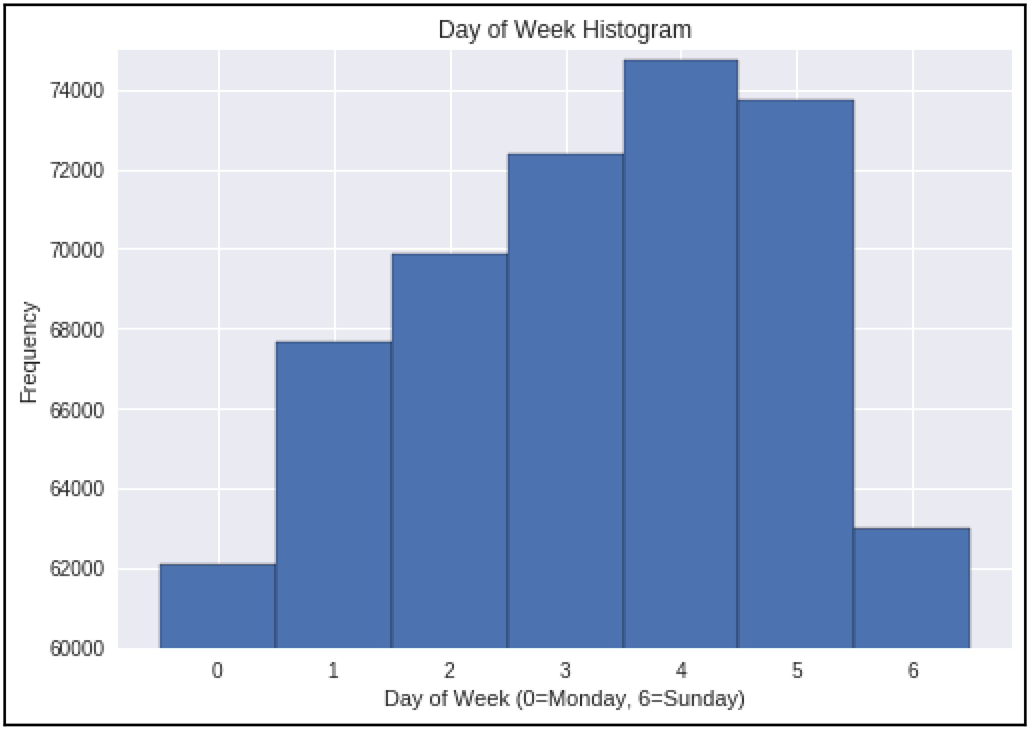
	key	fare_amount	pickup_datetime	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
0	2009-06-15 17:26:21.0000001	4.5	2009-06-15 17:26:21	-73.844311	40.721319	-73.841610	40.712278	1
1	2010-01-05 16:52:16.0000002	16.9	2010-01-05 16:52:16	-74.016048	40.711303	-73.979268	40.782004	1
2	2011-08-18 00:35:00.00000049	5.7	2011-08-18 00:35:00	-73.982738	40.761270	-73.991242	40.750562	2
3	2012-04-21 04:30:42.0000001	7.7	2012-04-21 04:30:42	-73.987130	40.733143	-73.991567	40.758092	1
4	2010-03-09 07:51:00.000000135	5.3	2010-03-09 07:51:00	-73.968095	40.768008	-73.956655	40.783762	1

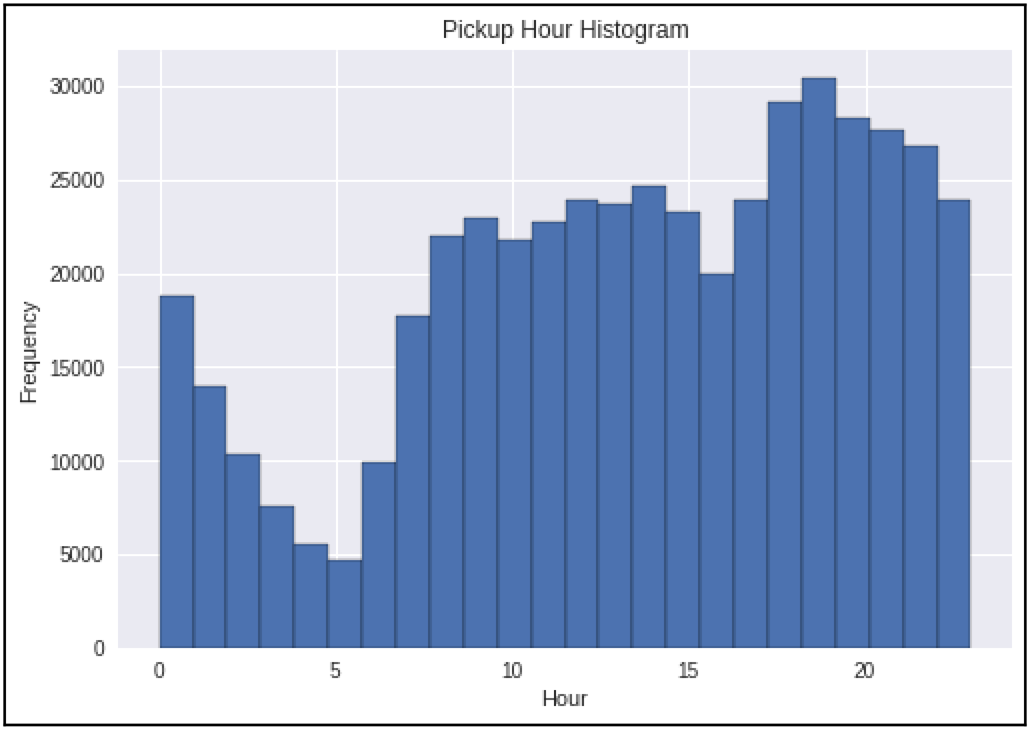




Pickup Locations in NYC Illustrated
40.85
40.80
40.75
40.70
40.65
Longitude
Upper Manhattan
Laguardia Airport
Midtown
Lower Manhattan
JFK Airport
-74.05
-74.00
-73.95
-73.90
-73.85
-73.80
-73.75
Latitude

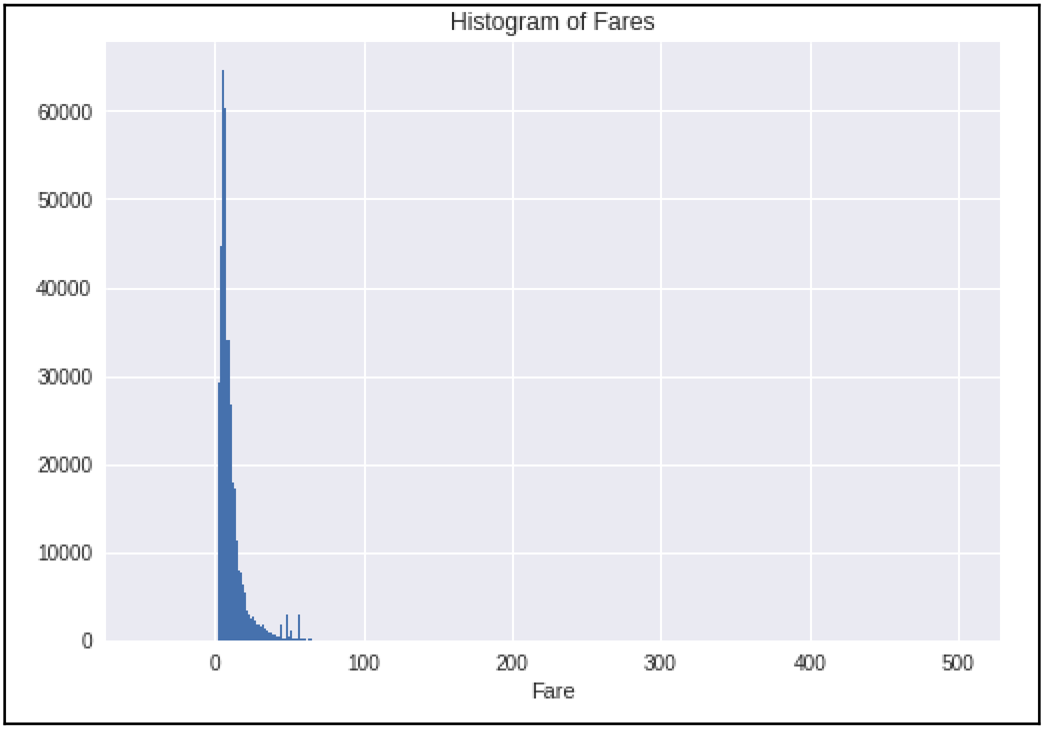

Dropoff Locations in NYC Illustrated
40.85
40.80
40.75
40.70
40.65
Longitude
-74.05
-74.00
-73.95
-73.90
-73.85
-73.80
-73.75
Latitude
Upper Manhattan
Laguardia Airport
Midtown
Lower Manhattan
Brooklyn
JFK Airport

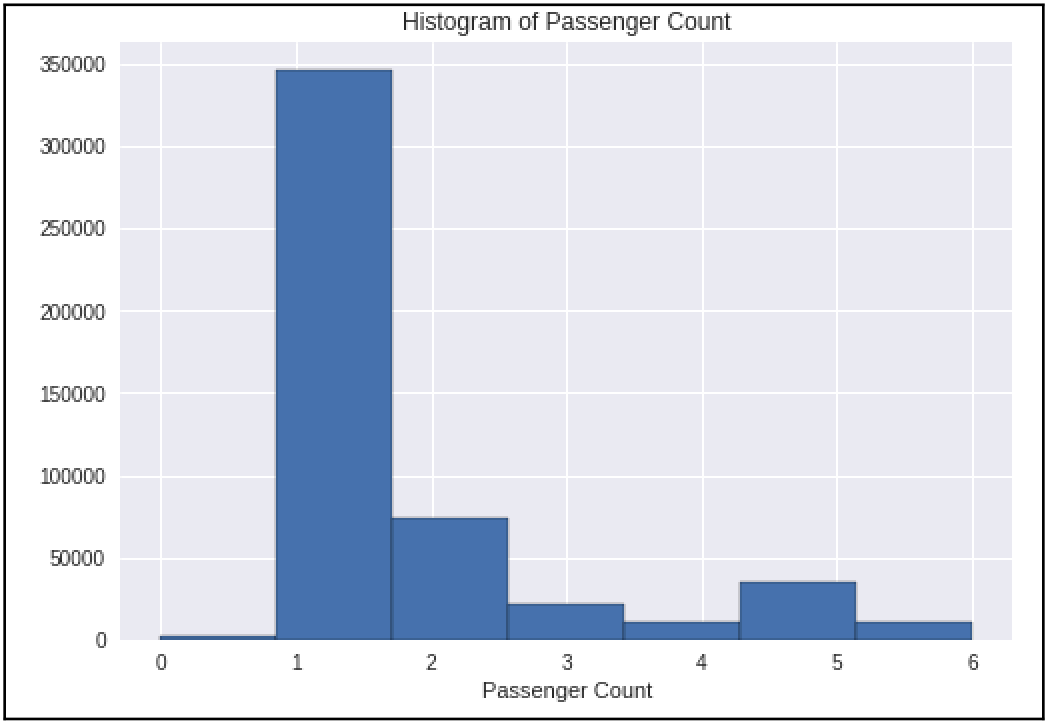

Day of Week Histogram
74000
72000
70000
68000
66000
64000
62000
60000
Frequency
0
1
2
3
4
5
6
Day of Week (0=Monday, 6=Sunday)

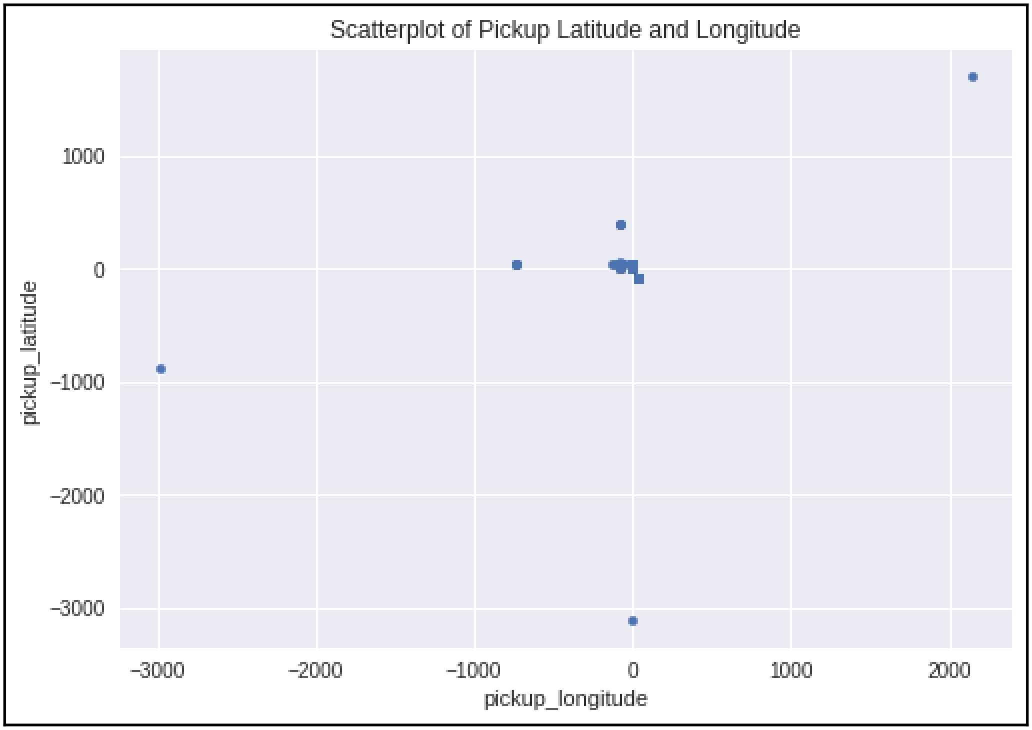



```
key                   0
fare_amount           0
pickup_datetime       0
pickup_longitude      0
pickup_latitude       0
dropoff_longitude     5
dropoff_latitude      5
passenger_count       0
dtype: int64
```

	fare_amount	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	passenger_count
count	499995.000000	499995.000000	499995.000000	499995.000000	499995.000000	499995.000000
mean	11.358182	-72.520091	39.920350	-72.522435	39.916526	1.683445
std	9.916069	11.856446	8.073318	11.797362	7.391002	1.307391
min	-44.900000	-2986.242495	-3116.285383	-3383.296608	-2559.748913	0.000000
25%	6.000000	-73.992047	40.734916	-73.991382	40.734057	1.000000
50%	8.500000	-73.981785	40.752670	-73.980126	40.753152	1.000000
75%	12.500000	-73.967117	40.767076	-73.963572	40.768135	2.000000
max	500.000000	2140.601160	1703.092772	40.851027	404.616667	6.000000



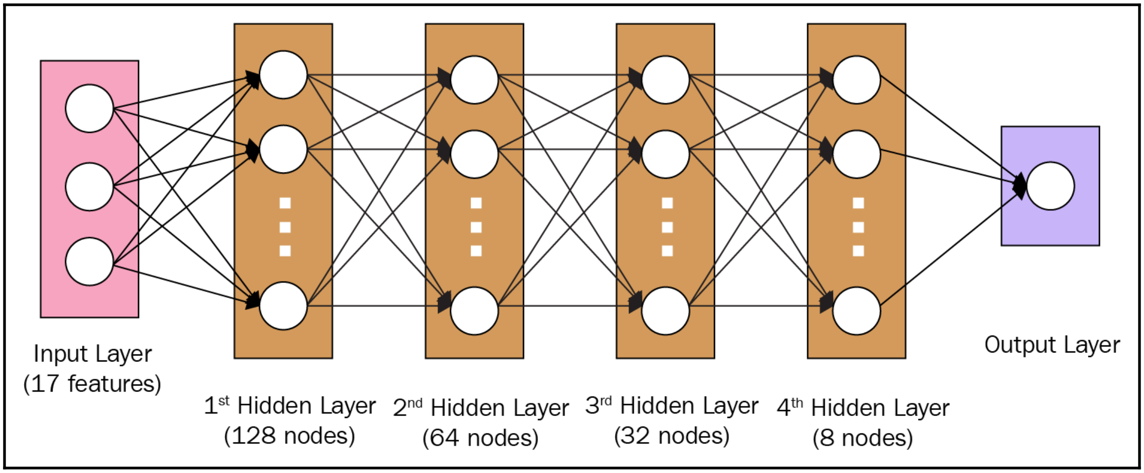

Histogram of Passenger Count
350000
300000
250000
200000
150000
100000
50000
0
0
1
2
3
4
5
6
Passenger Count




```
0    2009-06-15 17:26:21
1    2010-01-05 16:52:16
2    2011-08-18 00:35:00
3    2012-04-21 04:30:42
4    2010-03-09 07:51:00
Name: pickup_datetime, dtype: datetime64[ns]
```

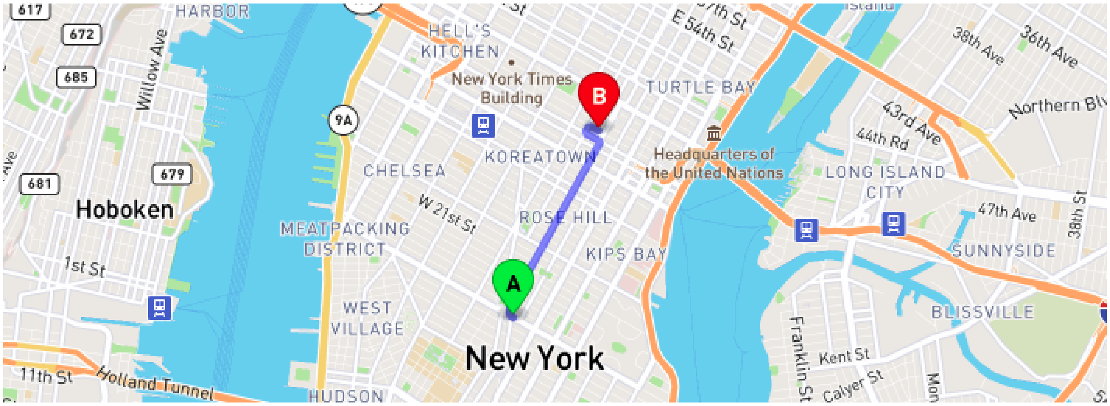
	pickup_datetime	year	month	day	day_of_week	hour
0	2009-06-15 17:26:21	2009	6	15	0	17
1	2010-01-05 16:52:16	2010	1	5	1	16
2	2011-08-18 00:35:00	2011	8	18	3	0
3	2012-04-21 04:30:42	2012	4	21	5	4
4	2010-03-09 07:51:00	2010	3	9	1	7
5	2011-01-06 09:50:45	2011	1	6	3	9

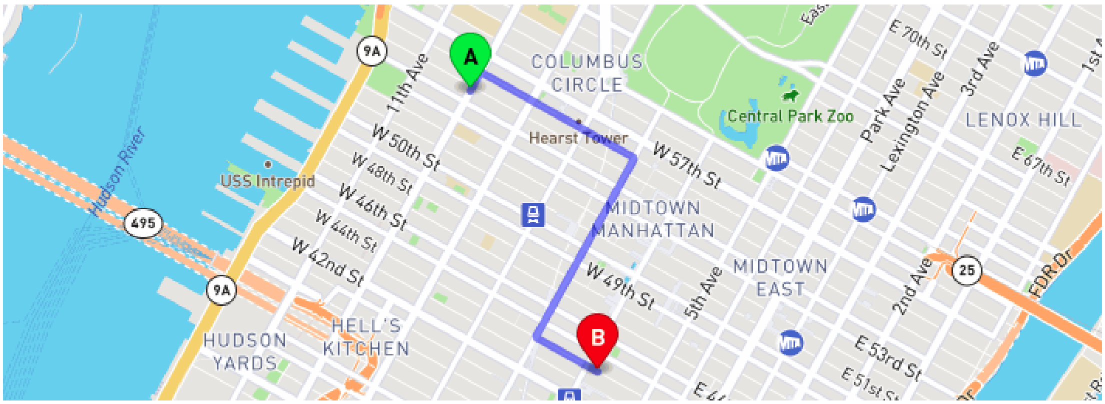
	key	pickup_longitude	pickup_latitude	dropoff_longitude	dropoff_latitude	pickup_dist_JFK_Airport	dropoff_dist_JFK_Airport
0	2009-06-15 17:26:21.0000001	-73.844311	40.721319	-73.841610	40.712278	0.101340	0.092710
1	2010-01-05 16:52:16.0000002	-74.016048	40.711303	-73.979268	40.782004	0.245731	0.242961
2	2011-08-18 00:35:00.00000049	-73.982738	40.761270	-73.991242	40.750562	0.234714	0.237050
3	2012-04-21 04:30:42.0000001	-73.987130	40.733143	-73.991567	40.758092	0.225895	0.240846
4	2010-03-09 07:51:00.000000135	-73.968095	40.768008	-73.956655	40.783762	0.225847	0.225878

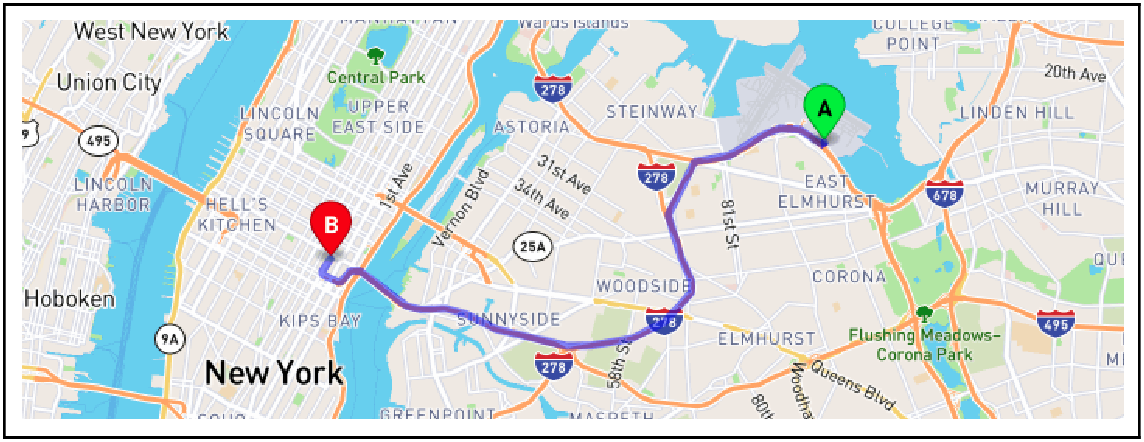


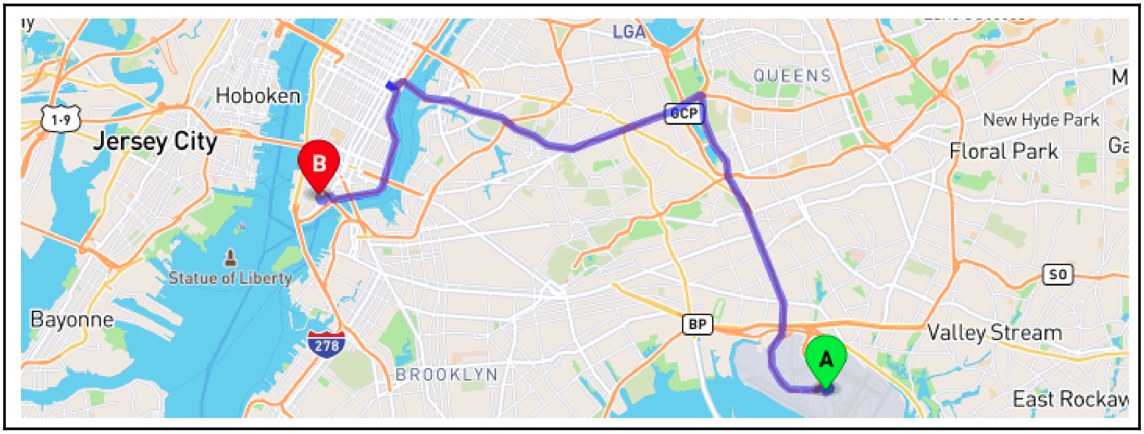

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 128)               2304
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_4 (Dense)              (None, 8)                 264
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 9
=================================================================
Total params: 12,913
Trainable params: 12,913
Non-trainable params: 0
_________________________________________________________________
```

```
Epoch 1/1
386741/386741 [==============================] - 106s 275us/step - loss: 15.4968 - mean_squared_error: 15.4968
<keras.callbacks.History at 0x7fef288532b0>
```



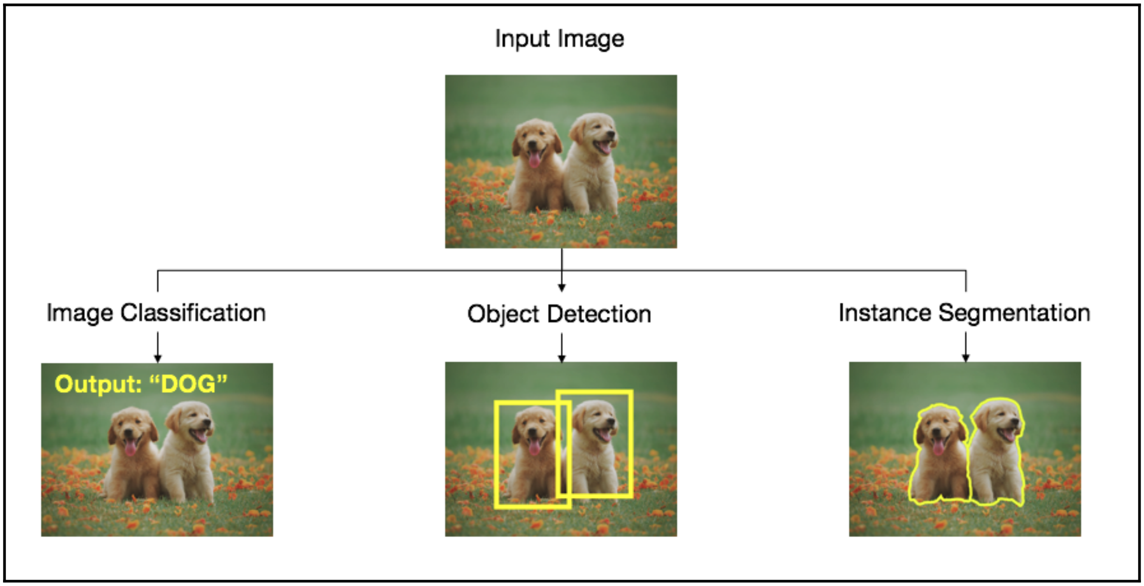


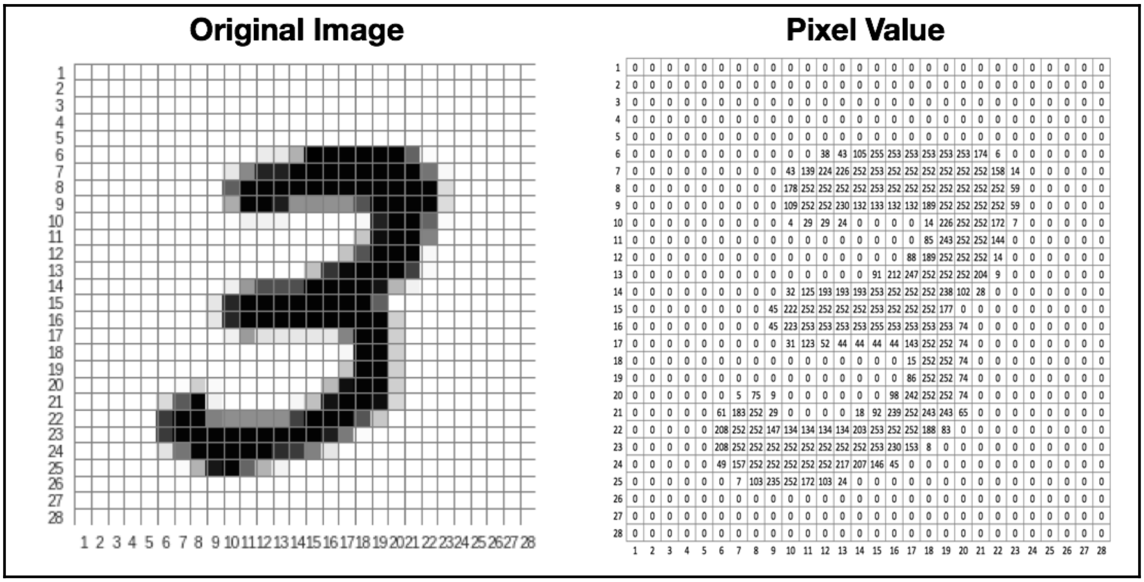


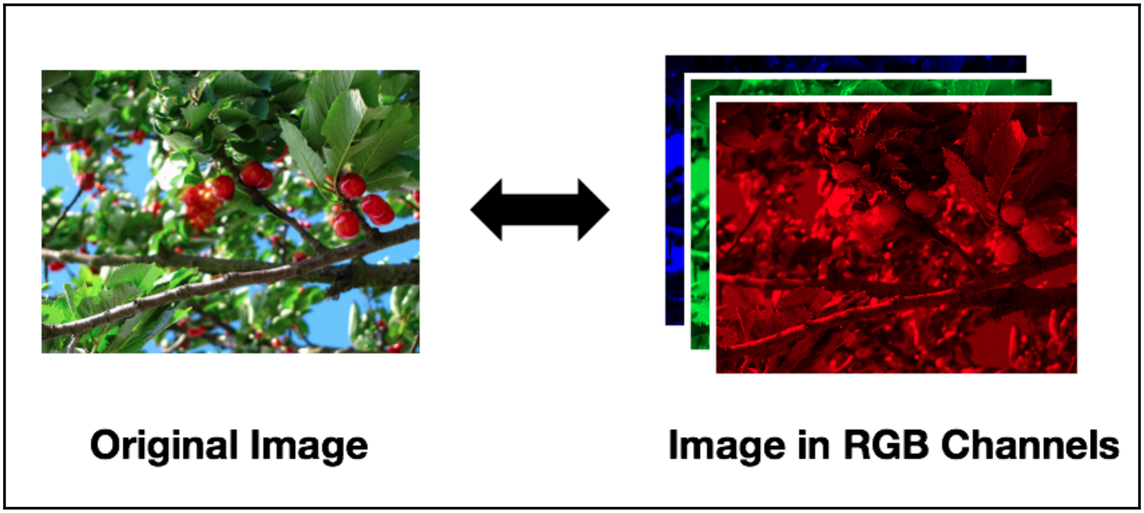


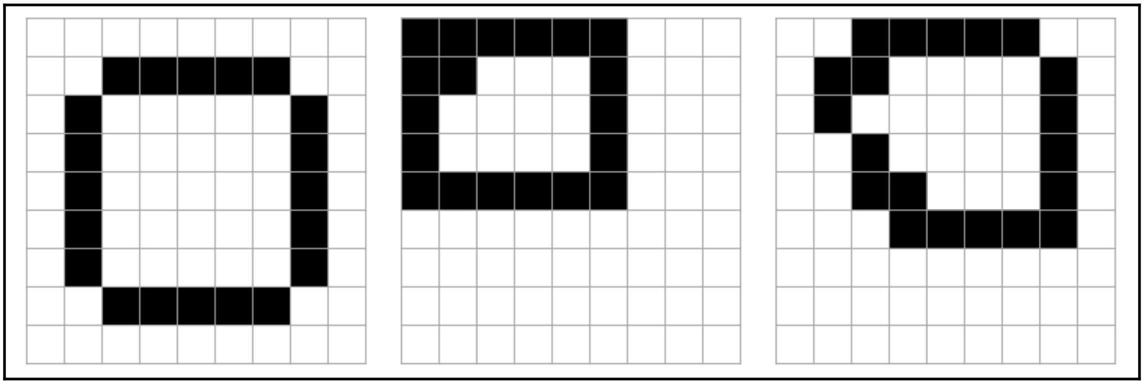

```
Train RMSE: 3.52
Test RMSE: 3.55
```

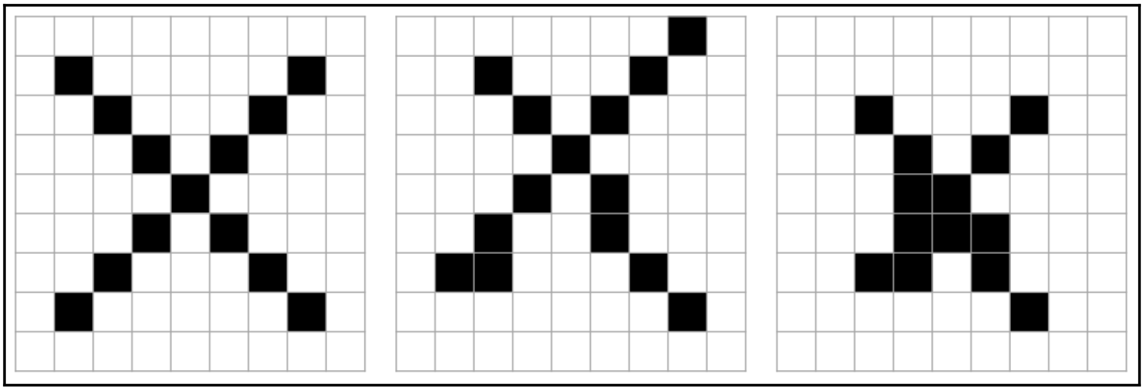
Chapter 4: Cats Versus Dogs - Image Classification Using CNNs

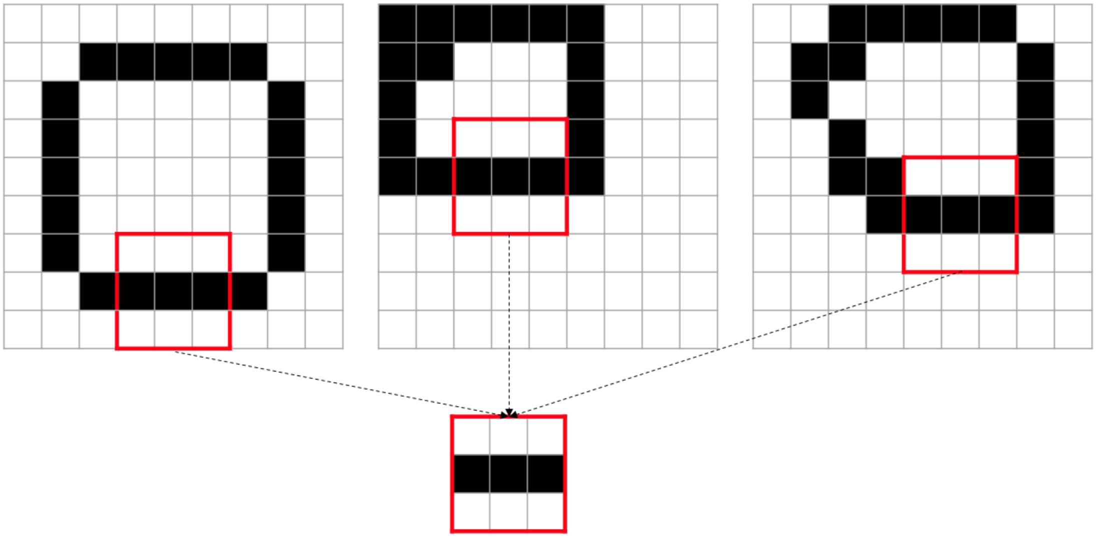




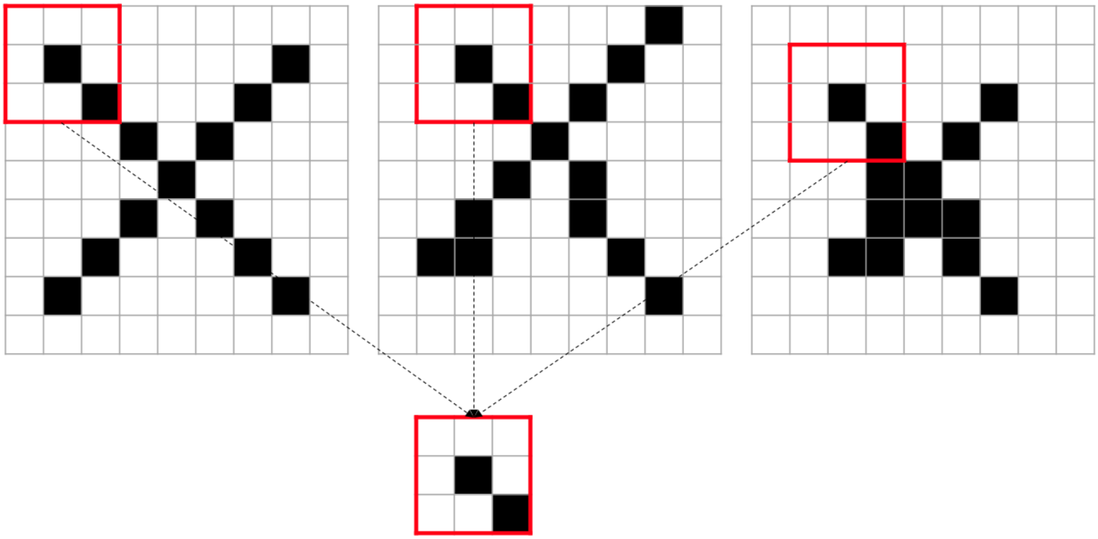

Original Image
Image in RGB Channels



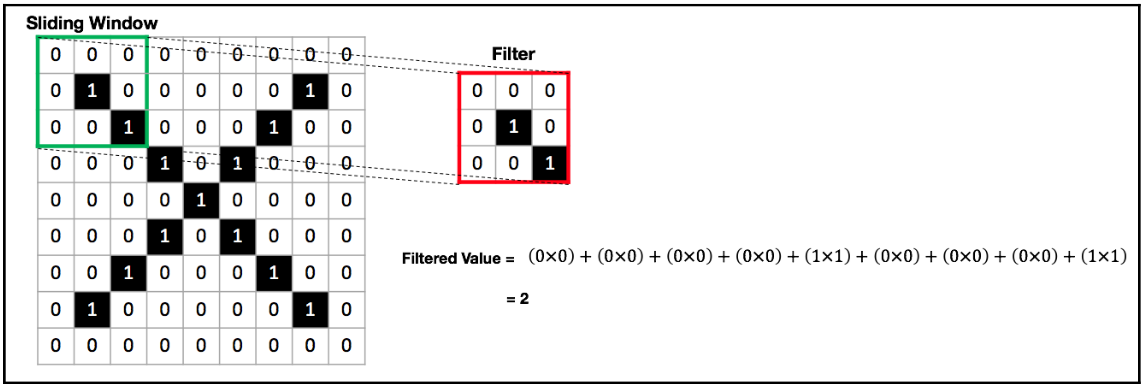


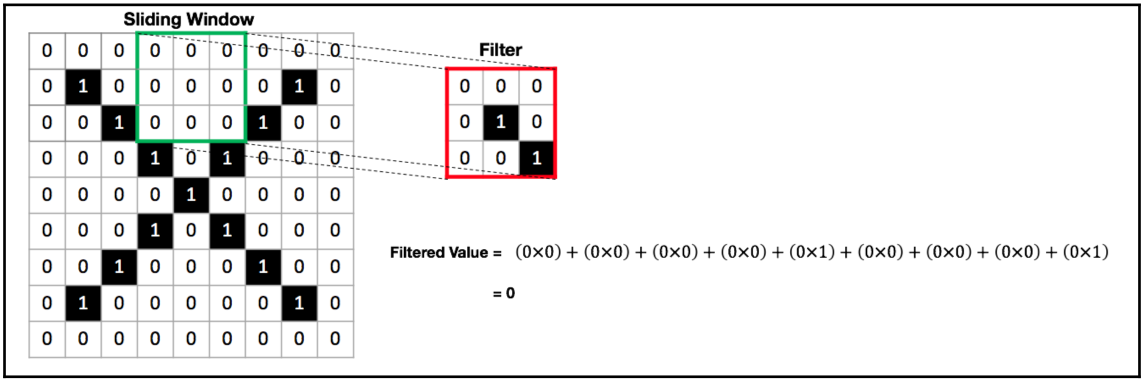


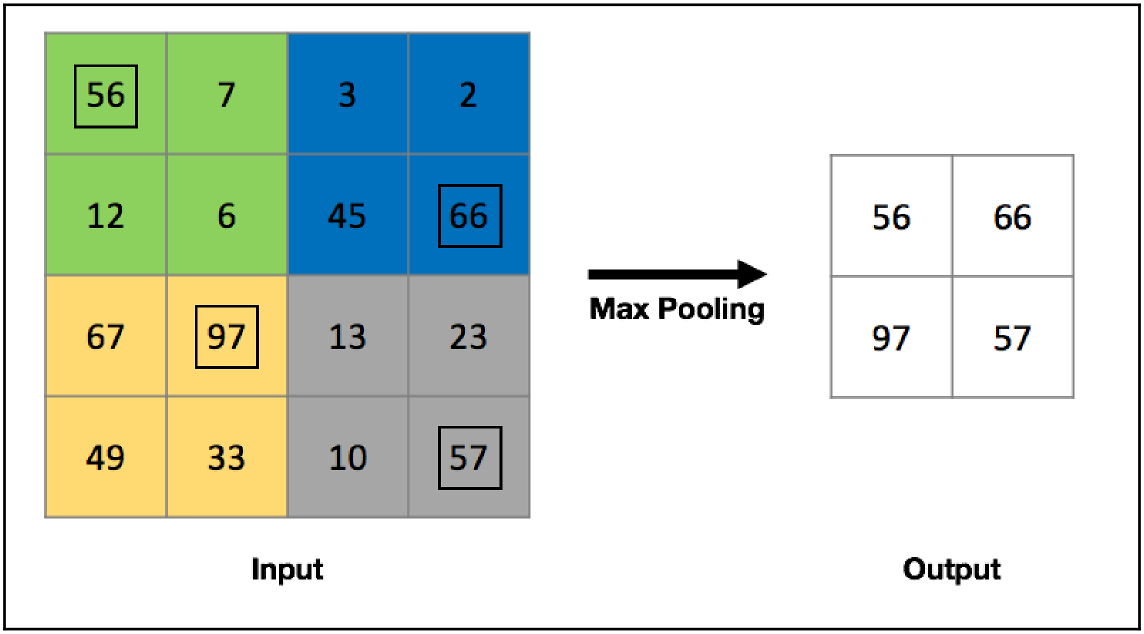
Characteristic Feature for ‘O’
(also known as a filter)

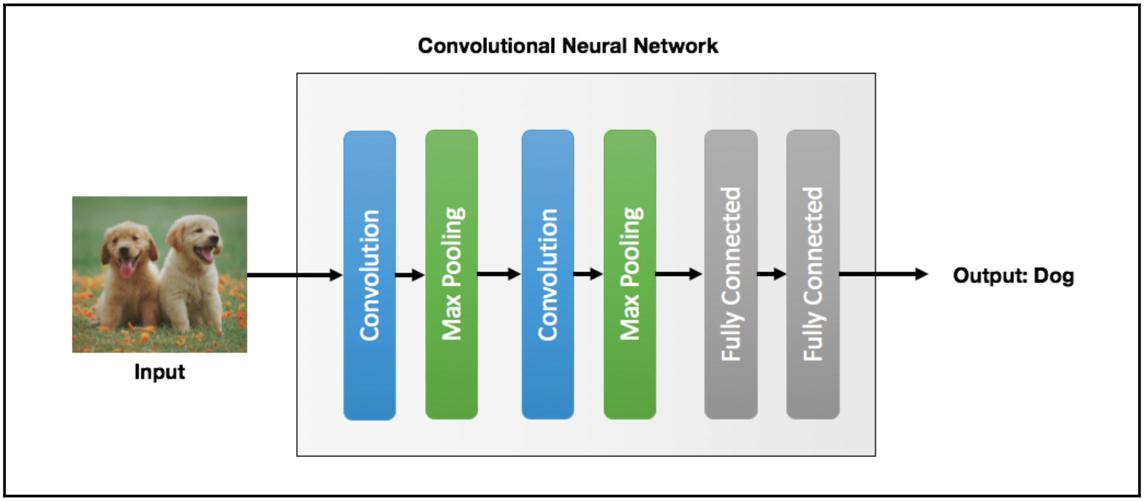


Characteristic Feature for ‘X’
(also known as a filter)


Sliding Window
0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0
Filter
0 0 0
0 1 0
0 0 1
Filtered Value = (0×0) + (0×0) + (0×0) + (0×0) + (1×1) + (0×0) + (0×0) + (0×0) + (1×1)
= 2


Sliding Window
0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 0
0 0 1 0 0 0 1 0 0
0 0 0 1 0 1 0 0 0
0 0 0 0 1 0 0 0 0
0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 1 0 0
0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0
Filter
0 0 0
0 1 0
0 0 1
Filtered Value = (0×0) + (0×0) + (0×0) + (0×0) + (0×1) + (0×0) + (0×0) + (0×0) + (0×1)
= 0

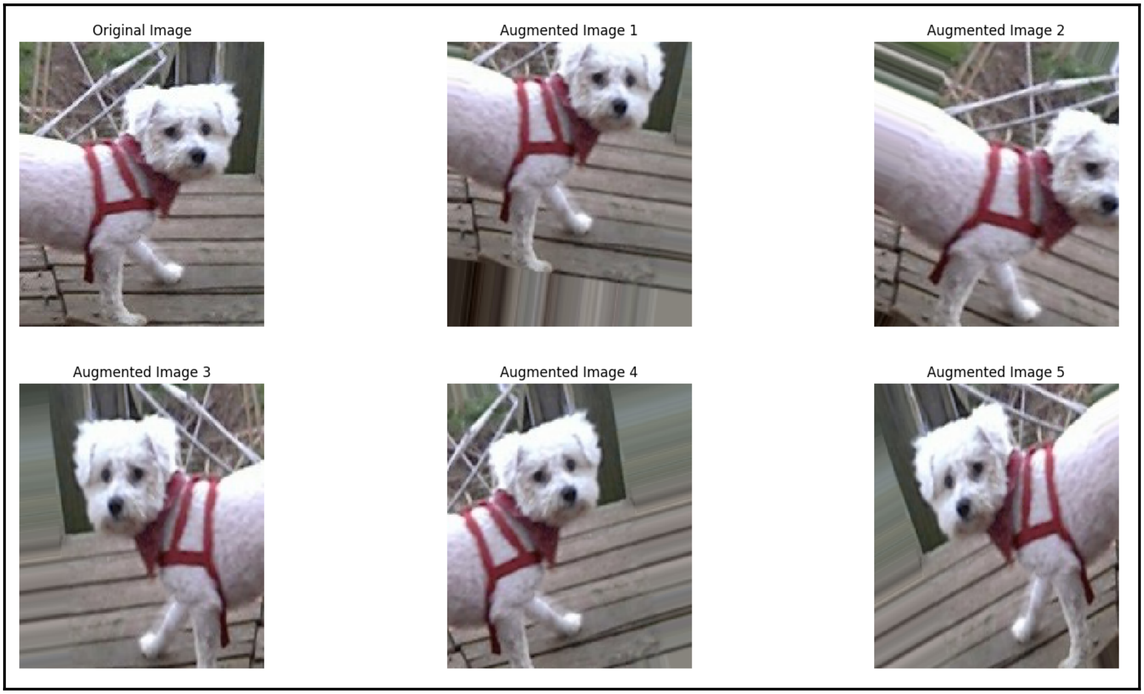

56
7
3
2
12
6
45
66
67
97
13
23
49
33
10
57
Max Pooling
56
66
97
57
Input
Output

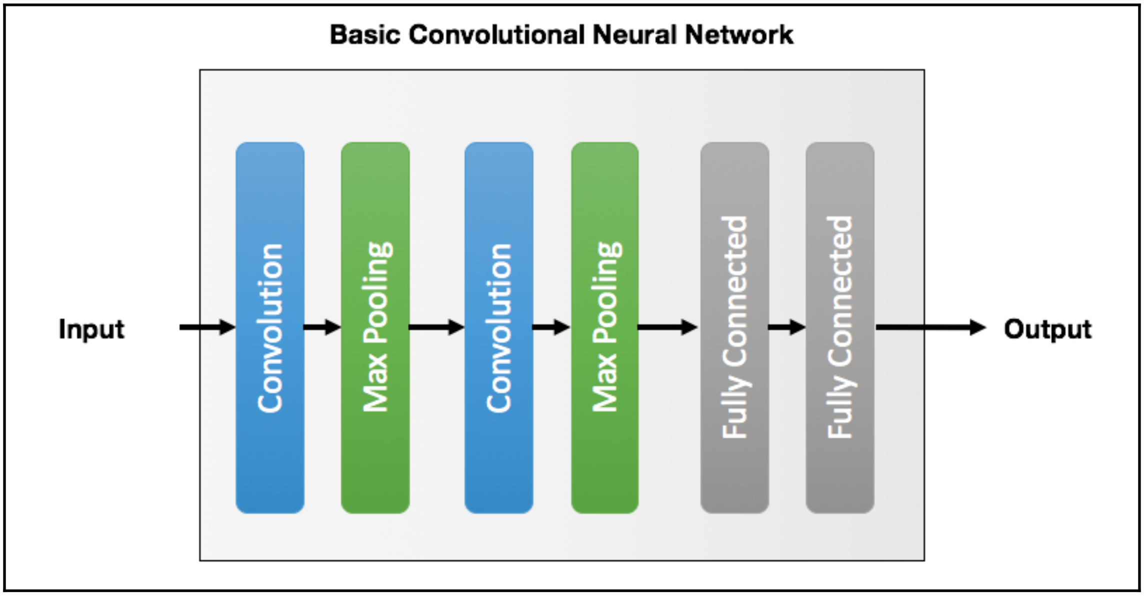

Convolutional Neural Network
Input
Convolution
Max Pooling
Convolution
Max Pooling
Fully Connected
Fully Connected
Output: Dog

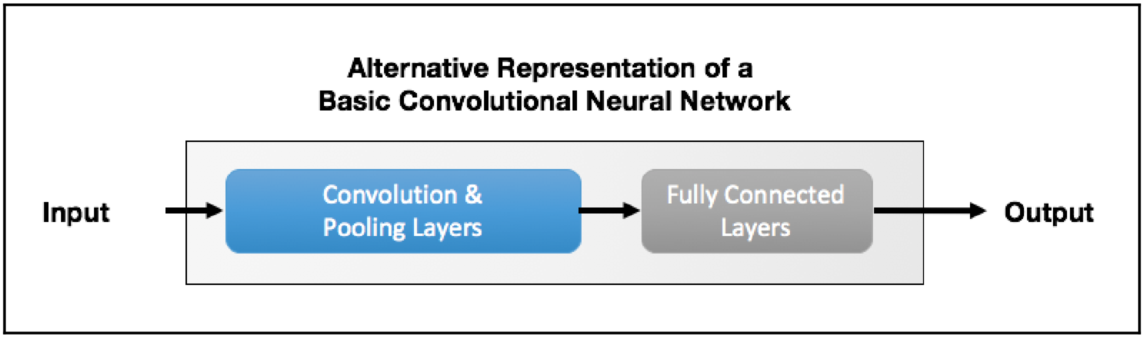

Cat/400.jpg
Cat/1079.jpg
Cat/3508.jpg
Cat/5817.jpg
Cat/6162.jpg
Cat/7805.jpg
Cat/4494.jpg
Cat/3164.jpg
Cat/8922.jpg


Dog/192.jpg
Dog/7880.jpg
Dog/4975.jpg
Dog/7152.jpg
Dog/4428.jpg
Dog/1989.jpg
Dog/11921.jpg
Dog/6199.jpg
Dog/4985.jpg

```
/data
    … /train
        … /cat
            … 0.jpg
            … 1.jpg
            … 2.jpg
        … /dog
            … 0.jpg
            … 1.jpg
            … 2.jpg
    … /test
        … /cat
            … 0.jpg
            … 1.jpg
            … 2.jpg
        … /dog
            … 0.jpg
            … 1.jpg
            … 2.jpg
```

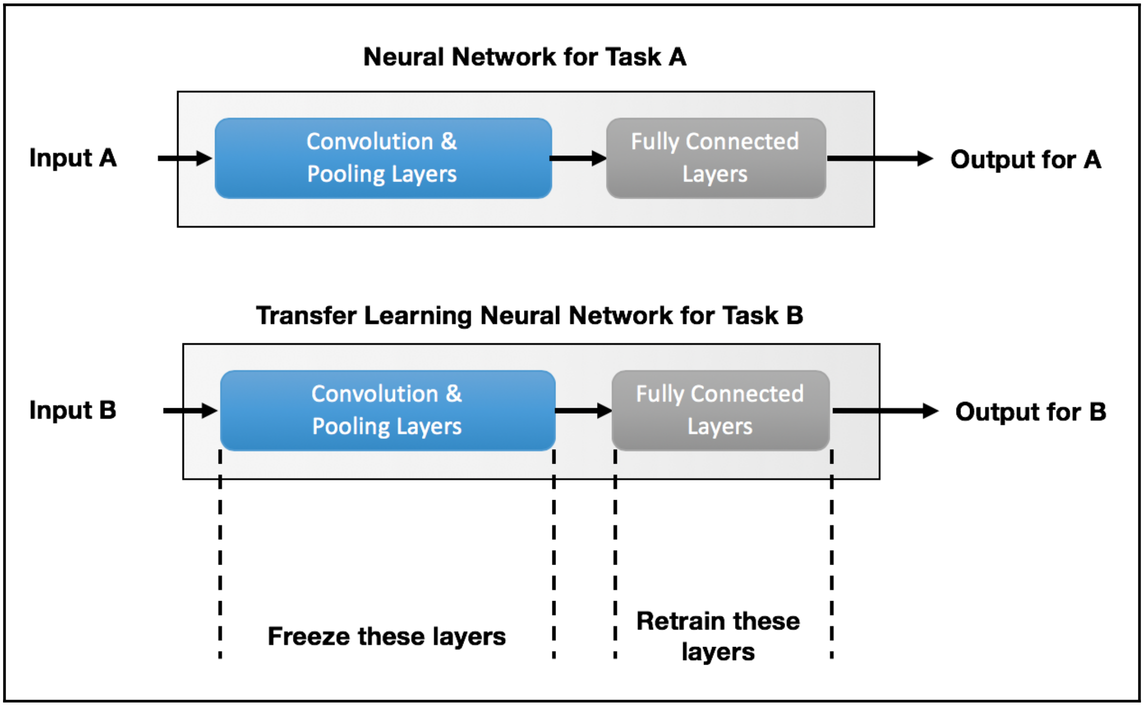

Original Image
Augmented Image 1
Augmented Image 2
Augmented Image 3
Augmented Image 4
Augmented Image 5


Basic Convolutional Neural Network
Input
Convolution
Max Pooling
Convolution
Max Pooling
Fully Connected
Fully Connected
Output




```
Epoch 1/10
1250/1250 [==============================] - 79s 63ms/step - loss: 0.6347 - acc: 0.6247
Epoch 2/10
1250/1250 [==============================] - 85s 68ms/step - loss: 0.5540 - acc: 0.7175
Epoch 3/10
1250/1250 [==============================] - 81s 65ms/step - loss: 0.5066 - acc: 0.7511
Epoch 4/10
1250/1250 [==============================] - 87s 69ms/step - loss: 0.4778 - acc: 0.7696
Epoch 5/10
1250/1250 [==============================] - 80s 64ms/step - loss: 0.4478 - acc: 0.7858
Epoch 6/10
1250/1250 [==============================] - 85s 68ms/step - loss: 0.4247 - acc: 0.8054
Epoch 7/10
1250/1250 [==============================] - 81s 65ms/step - loss: 0.4007 - acc: 0.8141
Epoch 8/10
1250/1250 [==============================] - 82s 65ms/step - loss: 0.3835 - acc: 0.8241
Epoch 9/10
1250/1250 [==============================] - 85s 68ms/step - loss: 0.3635 - acc: 0.8371
Epoch 10/10
1250/1250 [==============================] - 81s 65ms/step - loss: 0.3395 - acc: 0.8486
```

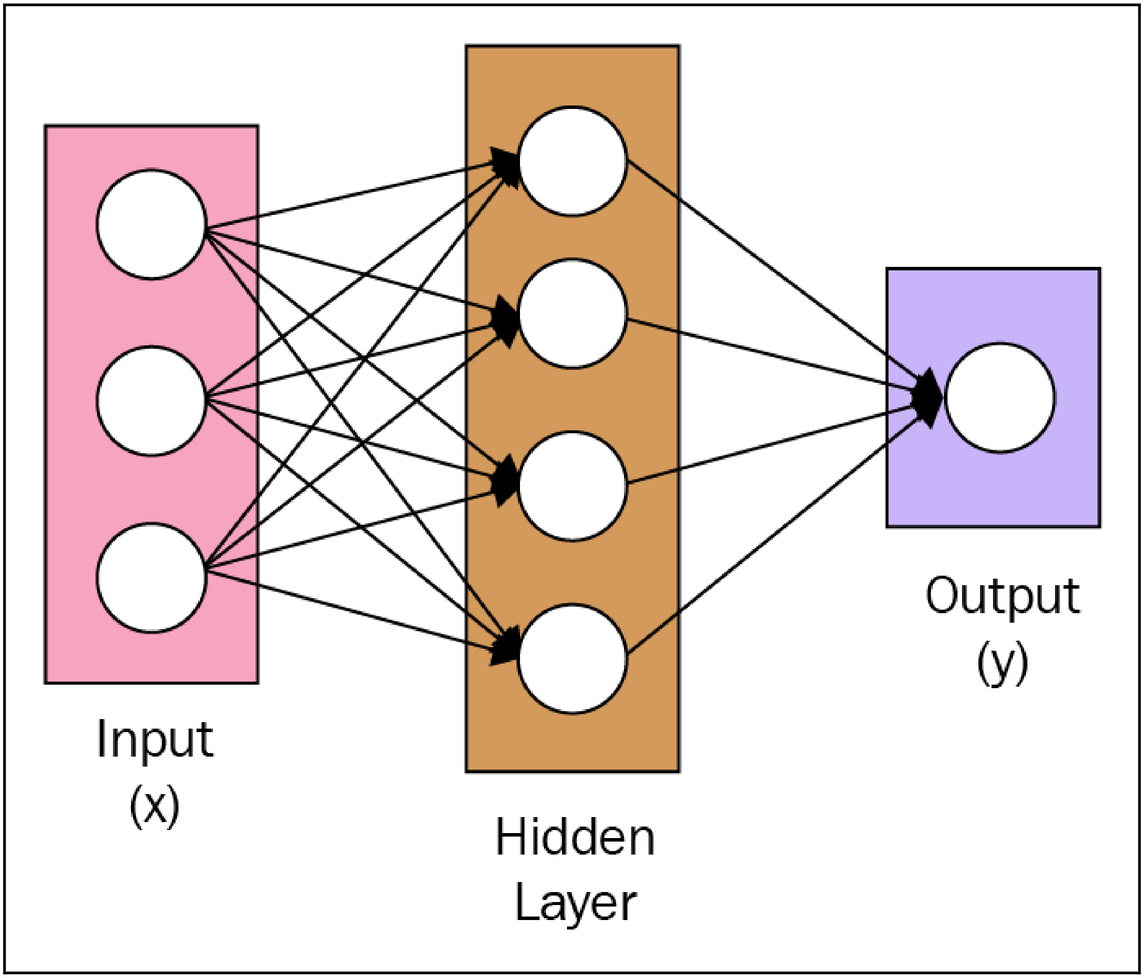
```
loss: 0.8116428855985403
acc: 0.8054
```

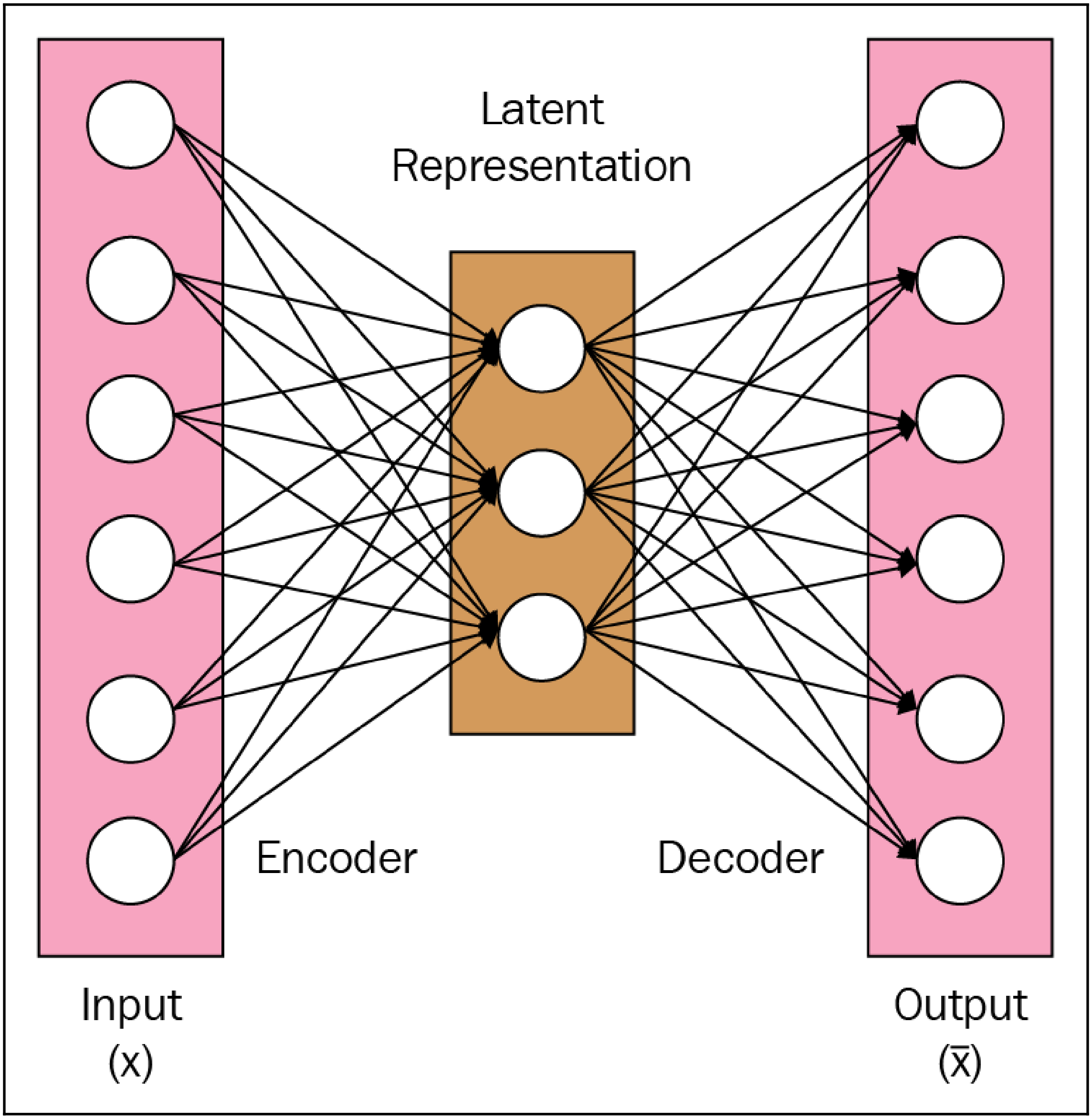


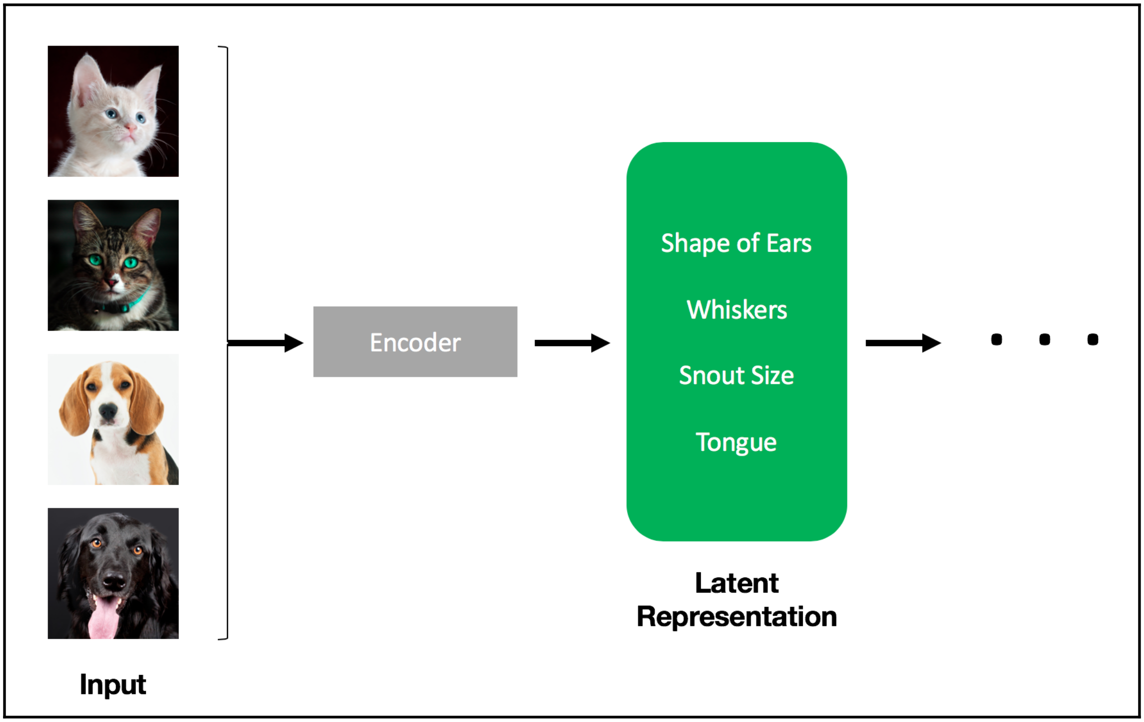

```
Epoch 1/3
200/200 [==============================] - 381s 2s/step - loss: 0.3808 - acc: 0.8253
Epoch 2/3
200/200 [==============================] - 418s 2s/step - loss: 0.2903 - acc: 0.8731
Epoch 3/3
200/200 [==============================] - 404s 2s/step - loss: 0.2941 - acc: 0.8754
```

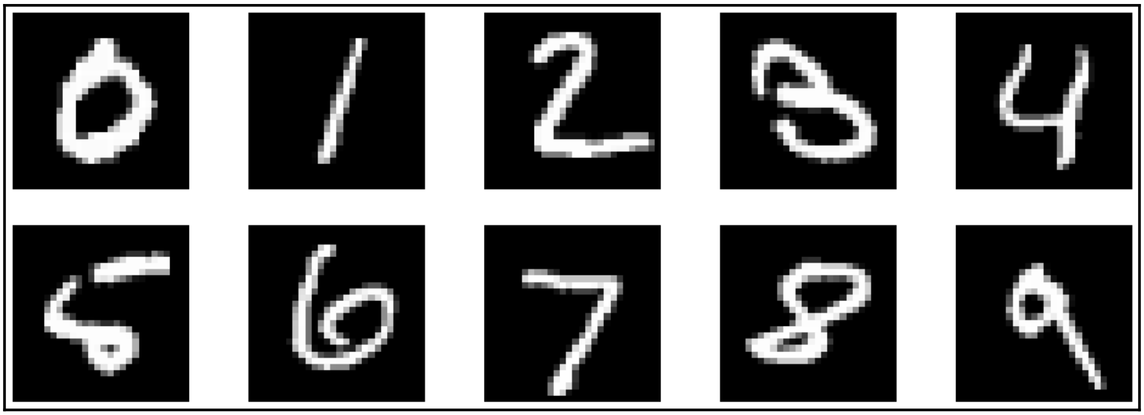
```
loss: 0.23026393374204635
acc: 0.905
```

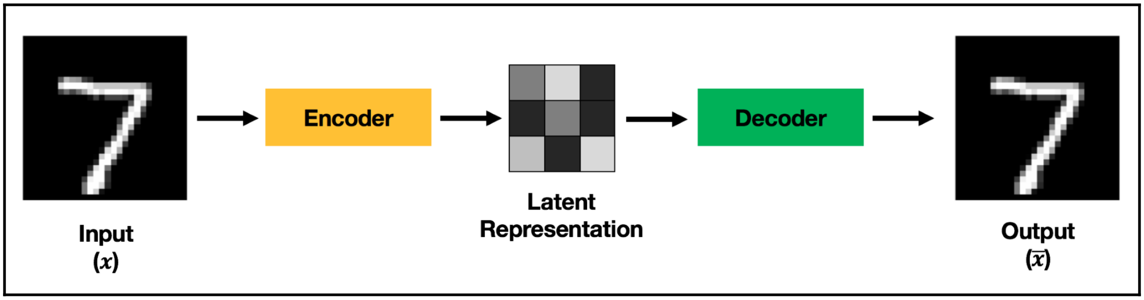
Chapter 5: Removing Noise from Images Using Autoencoders

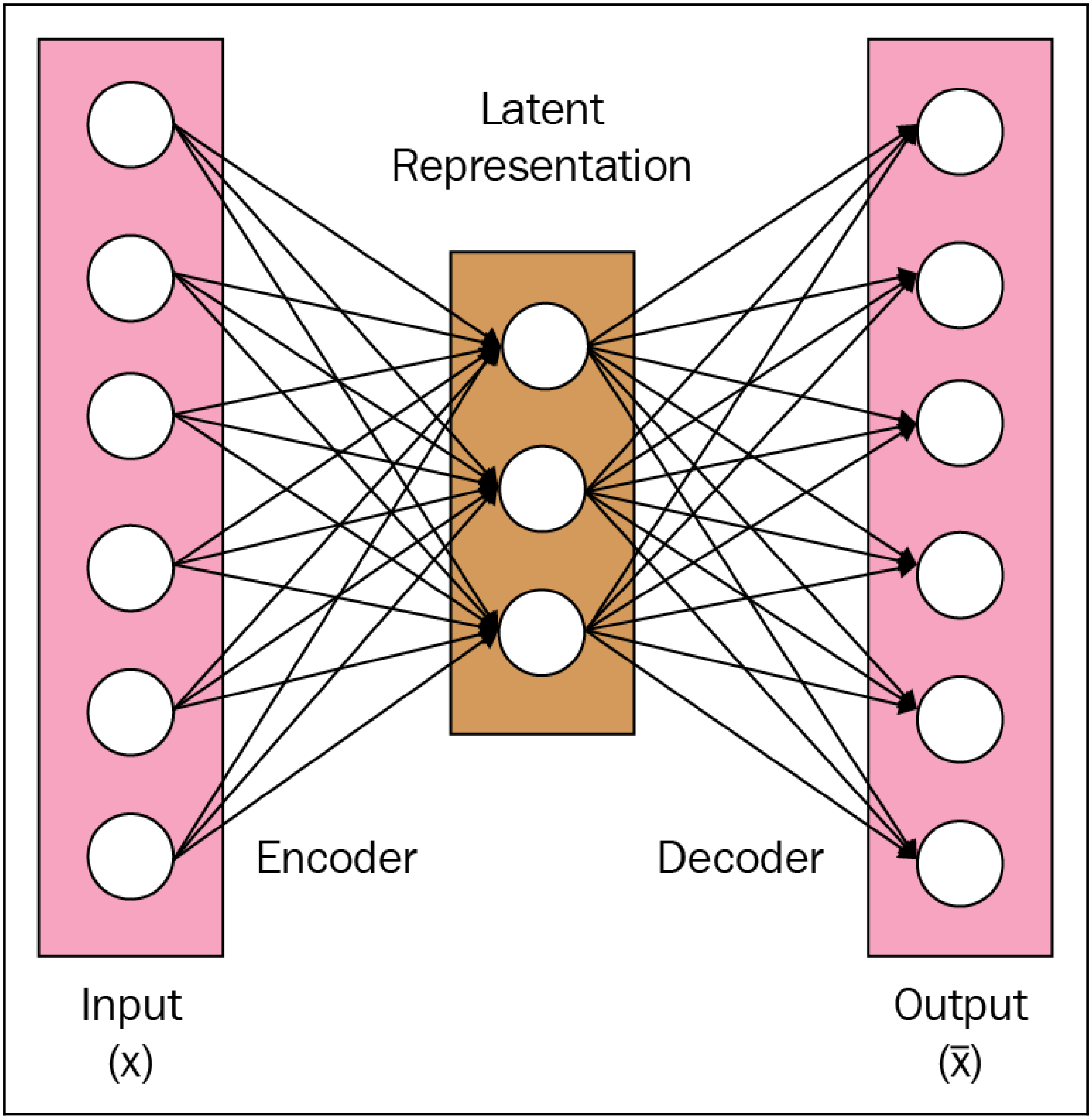



Latent
Representation
Encoder
Decoder
Input
(x)
Output
($\bar{x}$)


Encoder
Shape of Ears
Whiskers
Snout Size
Tongue
Latent Representation
Input

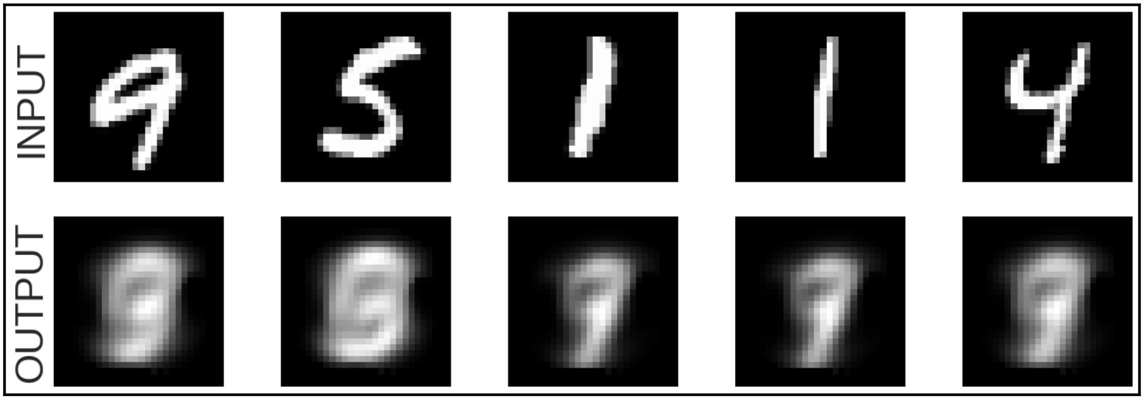


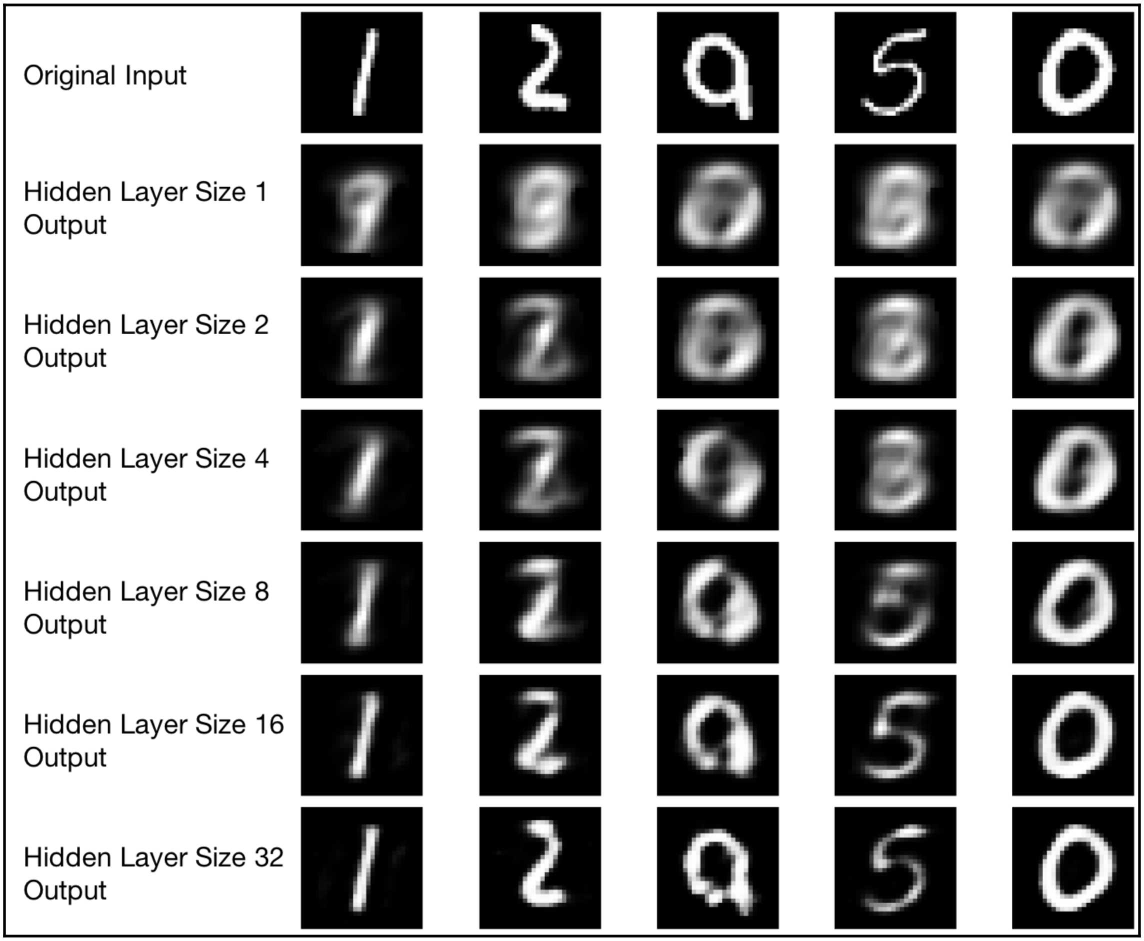

Encoder
Latent
Representation
Decoder
Input
(x)
Output
($\overline{x}$)

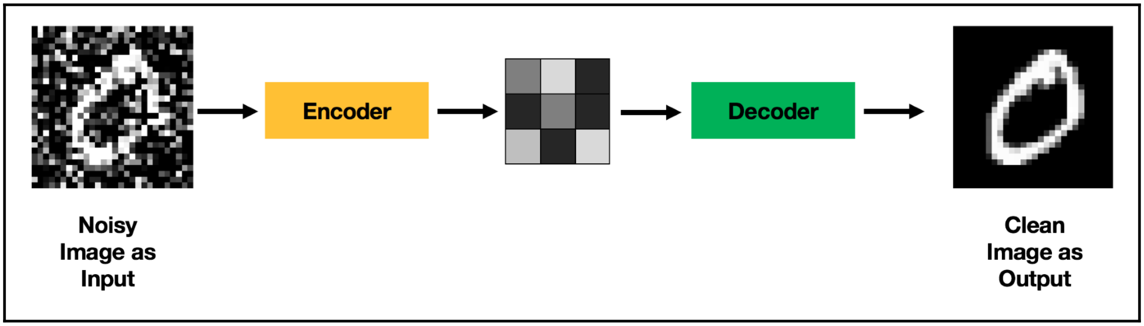

Latent
Representation
Encoder
Decoder
Input
(x)
Output
(x̄)

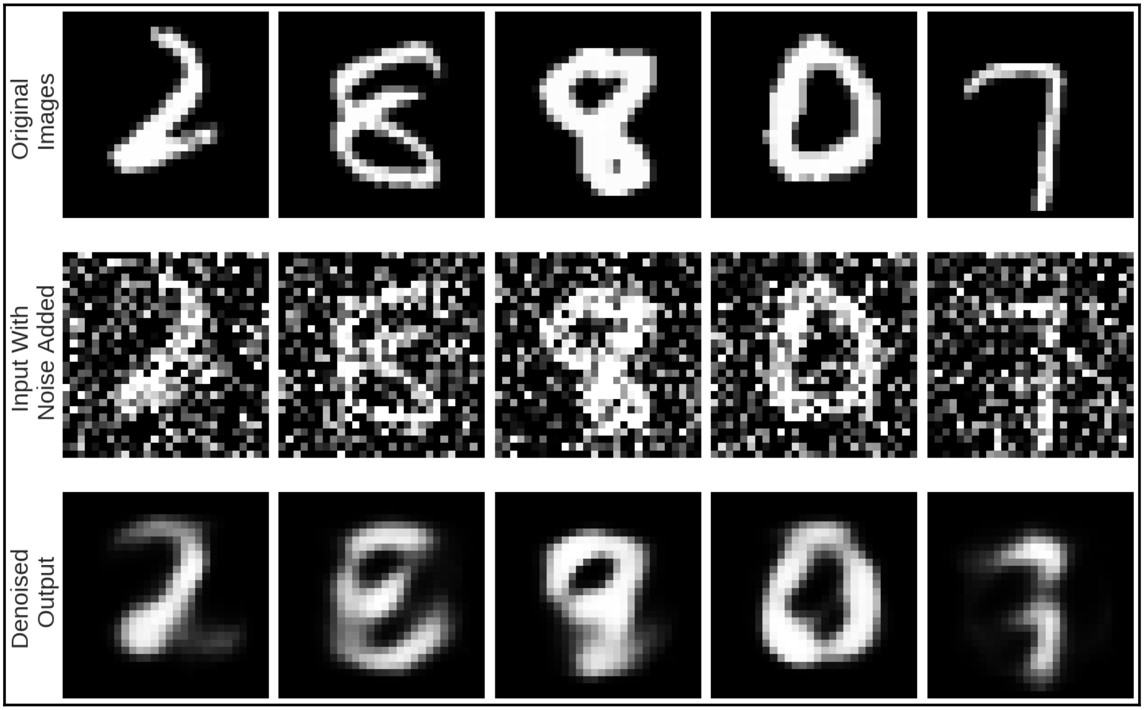
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 1)                 785
_________________________________________________________________
dense_2 (Dense)              (None, 784)               1568
=================================================================
Total params: 2,353
Trainable params: 2,353
Non-trainable params: 0
_________________________________________________________________
```

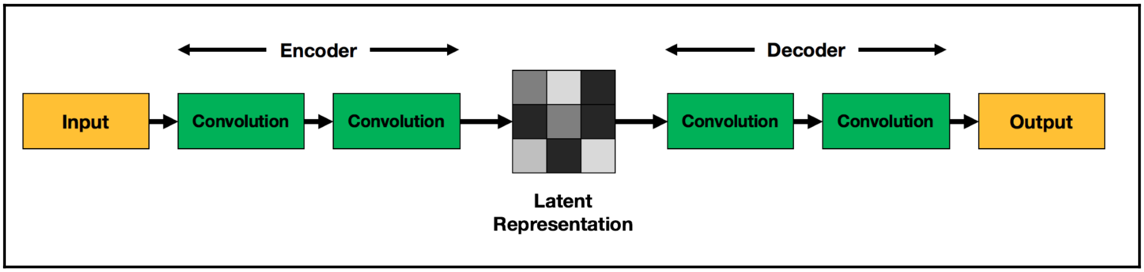
```
Epoch 1/10
60000/60000 [==============================] - 3s 51us/step - loss: 0.0750
Epoch 2/10
60000/60000 [==============================] - 3s 43us/step - loss: 0.0653
Epoch 3/10
60000/60000 [==============================] - 3s 47us/step - loss: 0.0641
Epoch 4/10
60000/60000 [==============================] - 3s 47us/step - loss: 0.0635
Epoch 5/10
60000/60000 [==============================] - 3s 44us/step - loss: 0.0632
Epoch 6/10
60000/60000 [==============================] - 3s 44us/step - loss: 0.0629
Epoch 7/10
60000/60000 [==============================] - 3s 43us/step - loss: 0.0625
Epoch 8/10
60000/60000 [==============================] - 3s 43us/step - loss: 0.0620
Epoch 9/10
60000/60000 [==============================] - 3s 43us/step - loss: 0.0616
Epoch 10/10
60000/60000 [==============================] - 3s 43us/step - loss: 0.0613
<keras.callbacks.History at 0x7fe7b2cadb00>
```


INPUT
OUTPUT


Original Input
Hidden Layer Size 1
Output
Hidden Layer Size 2
Output
Hidden Layer Size 4
Output
Hidden Layer Size 8
Output
Hidden Layer Size 16
Output
Hidden Layer Size 32
Output

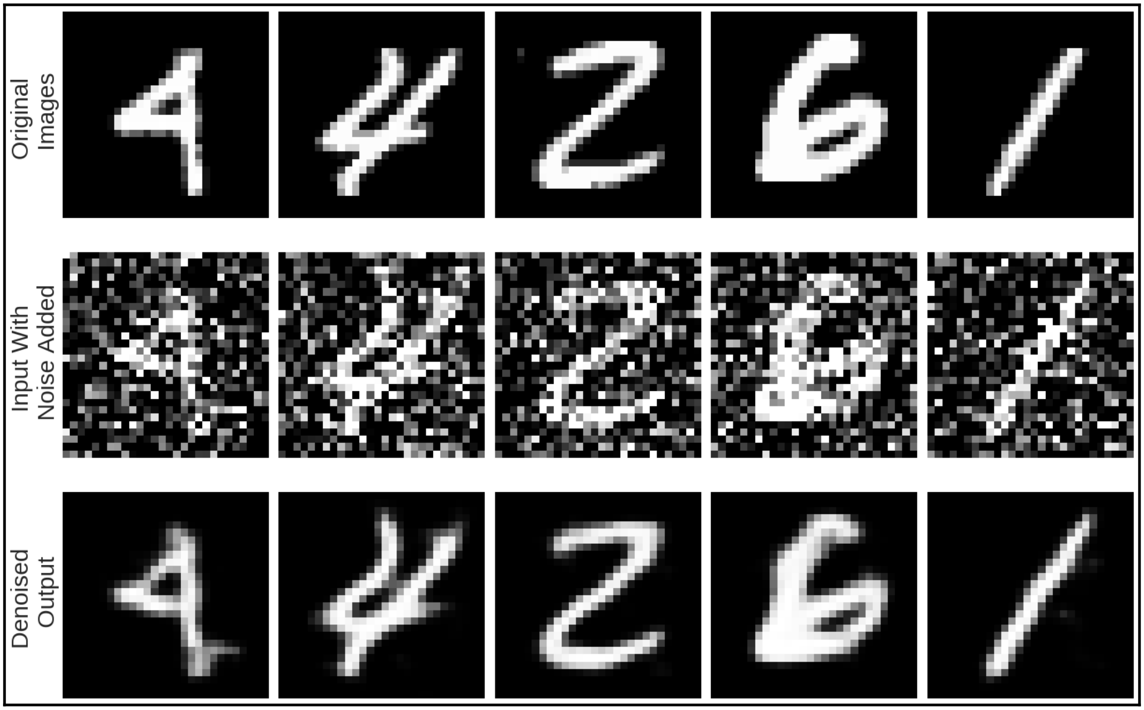

Encoder
Decoder
Noisy
Image as
Input
Clean
Image as
Output


Original
Images
Input With
Noise Added
Denoised
Output

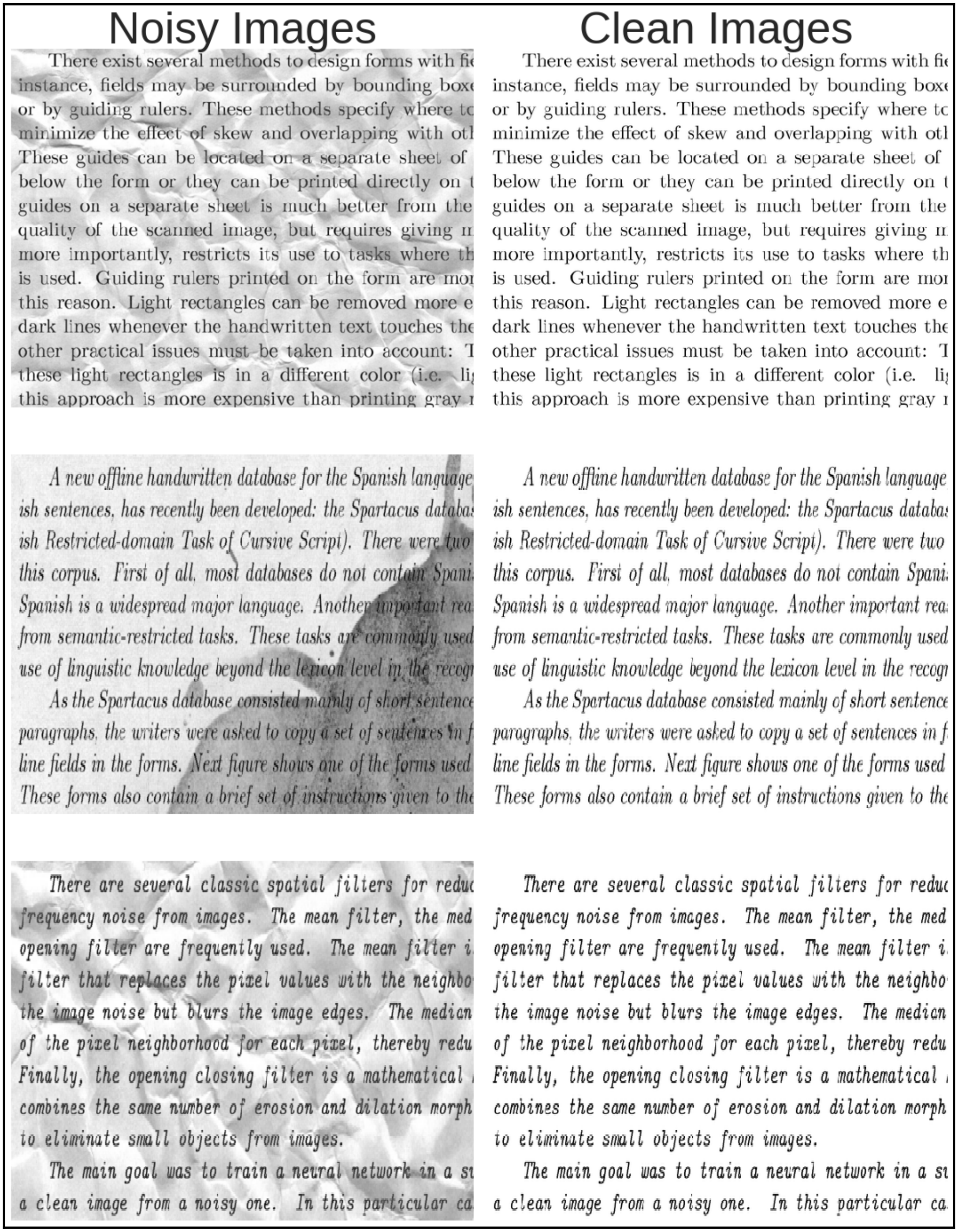

Encoder
Decoder
Input
Convolution
Convolution
Convolution
Convolution
Output
Latent
Representation

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 28, 28, 8)         1160
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 8)         584
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 28, 28, 16)        1168
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 28, 28, 1)         145
=================================================================
Total params: 3,217
Trainable params: 3,217
Non-trainable params: 0
```

```
Epoch 1/10
60000/60000 [==============================] - 17s 286us/step - loss: 0.1251
Epoch 2/10
60000/60000 [==============================] - 17s 279us/step - loss: 0.1039
Epoch 3/10
60000/60000 [==============================] - 17s 279us/step - loss: 0.1022
Epoch 4/10
60000/60000 [==============================] - 17s 280us/step - loss: 0.1012
Epoch 5/10
60000/60000 [==============================] - 17s 279us/step - loss: 0.1004
Epoch 6/10
60000/60000 [==============================] - 17s 279us/step - loss: 0.0998
Epoch 7/10
60000/60000 [==============================] - 17s 280us/step - loss: 0.0994
Epoch 8/10
60000/60000 [==============================] - 17s 280us/step - loss: 0.0990
Epoch 9/10
60000/60000 [==============================] - 17s 279us/step - loss: 0.0987
Epoch 10/10
60000/60000 [==============================] - 17s 282us/step - loss: 0.0985
```

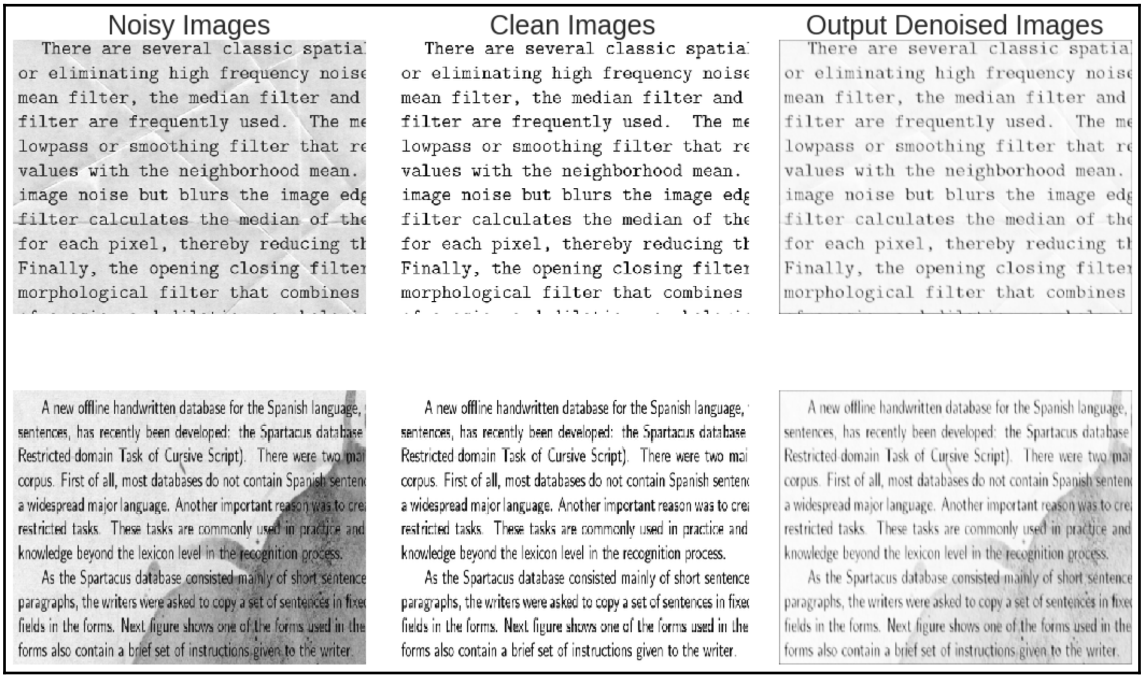



```
(216, 420, 540, 1)
```

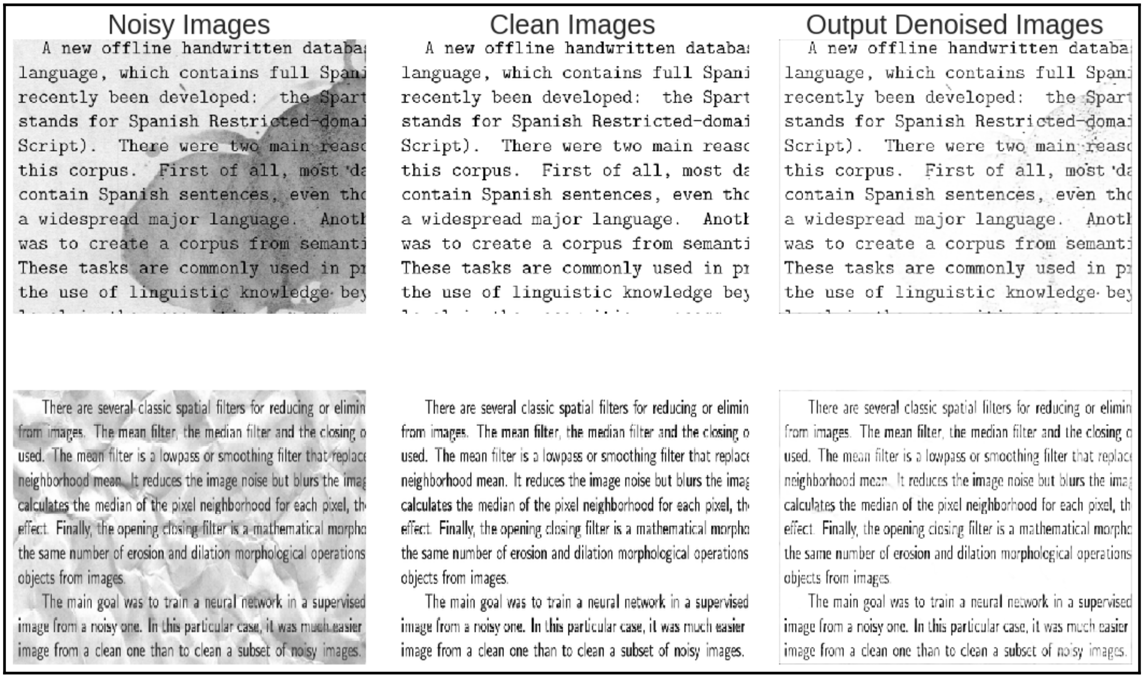


Noisy Images
Clean Images
There exist several methods to design forms with fi
instance, fields may be surrounded by bounding box
or by guiding rulers. These methods specify where to
minimize the effect of skew and overlapping with ot
These guides can be located on a separate sheet of
below the form or they can be printed directly on t
guides on a separate sheet is much better from the
quality of the scanned image, but requires giving m
more importantly, restricts its use to tasks where th
is used. Guiding rulers printed on the form are mor
this reason. Light rectangles can be removed more e
dark lines whenever the handwritten text touches the
other practical issues must be taken into account: T
these light rectangles is in a different color (i.e. li
this approach is more expensive than printing gray r
A new offline handwritten database for the Spanish language
ish sentences, has recently been developed: the Spartacus databa
ish Restricted-domain Task of Cursive Script). There were two
this corpus. First of all, most databases do not contain Spani
Spanish is a widespread major language. Another important rea
from semantic-restricted tasks. These tasks are commonly used
use of linguistic knowledge beyond the lexicon level in the recogn
As the Spartacus database consisted mainly of short sentence
paragraphs, the writers were asked to copy a set of sentences in f
line fields in the forms. Next figure shows one of the forms used
These forms also contain a brief set of instructions given to the
There are several classic spatial filters for reduc
frequency noise from images. The mean filter, the med
opening filter are frequently used. The mean filter i
filter that replaces the pixel values with the neighbo
the image noise but blurs the image edges. The median
of the pixel neighborhood for each pixel, thereby redu
Finally, the opening closing filter is a mathematical
combines the same number of erosion and dilation morph
to eliminate small objects from images.
The main goal was to train a neural network in a su
a clean image from a noisy one. In this particular ca

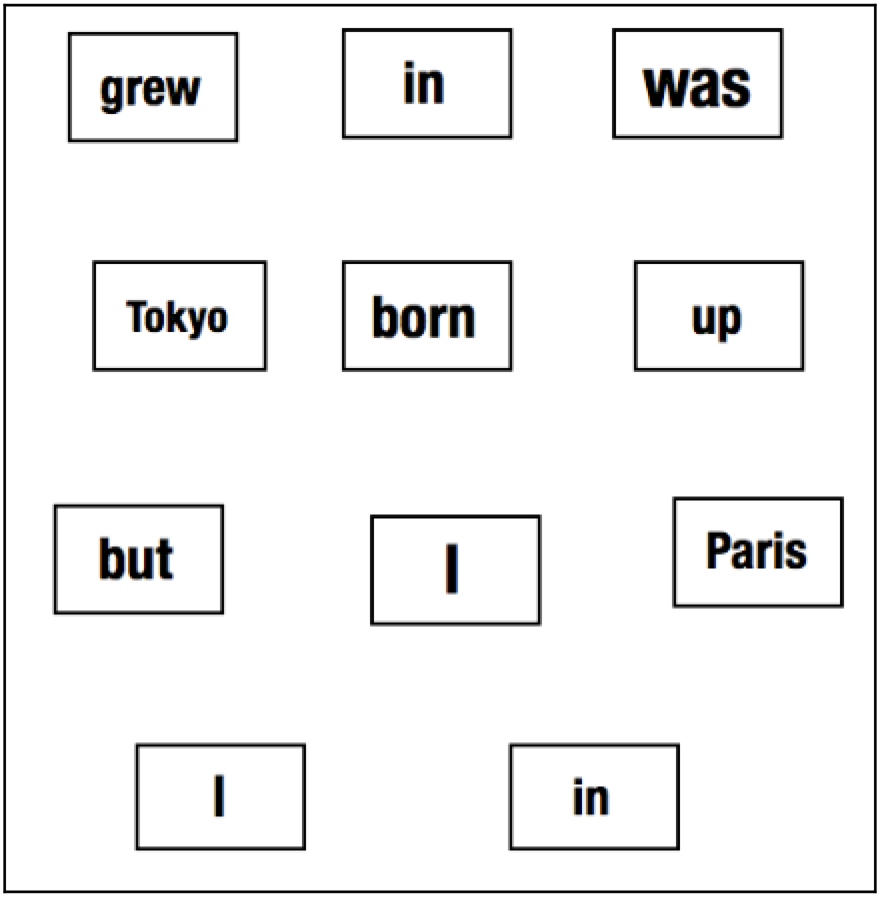
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_26 (Conv2D)           (None, 420, 540, 8)       80
_________________________________________________________________
conv2d_27 (Conv2D)           (None, 420, 540, 8)       584
_________________________________________________________________
conv2d_28 (Conv2D)           (None, 420, 540, 1)       73
=================================================================
Total params: 737
Trainable params: 737
Non-trainable params: 0
```

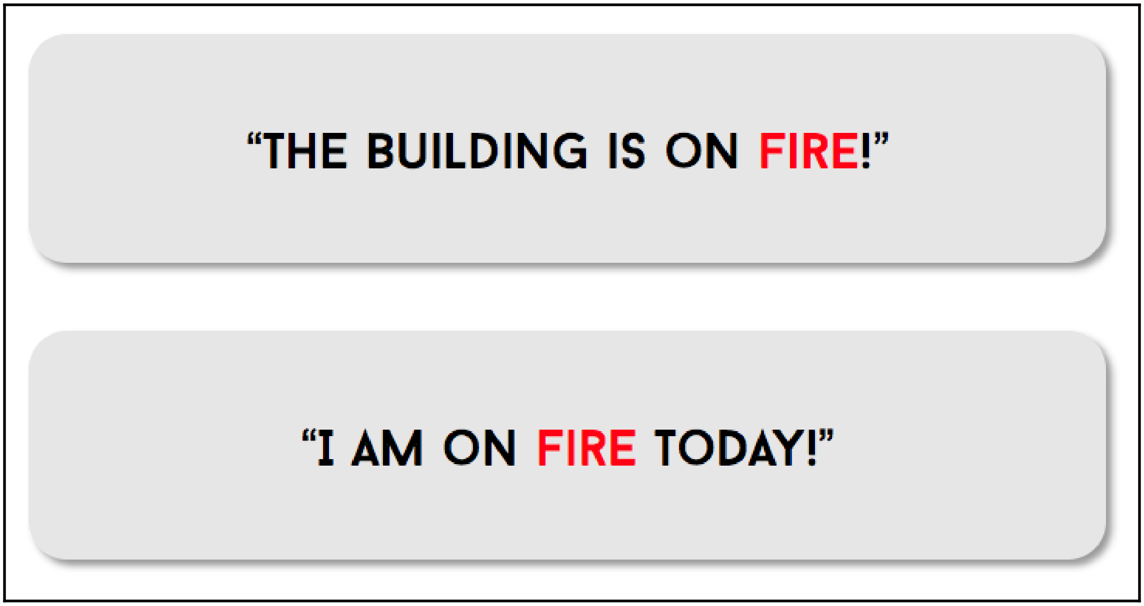


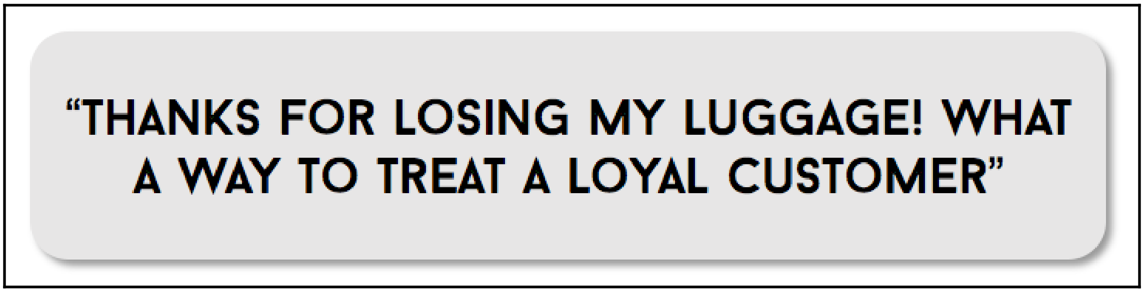
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_29 (Conv2D)           (None, 420, 540, 32)      320
_________________________________________________________________
conv2d_30 (Conv2D)           (None, 420, 540, 16)      4624
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 420, 540, 8)       1160
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 420, 540, 8)       584
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 420, 540, 16)      1168
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 420, 540, 32)      4640
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 420, 540, 1)       289
=================================================================
Total params: 12,785
Trainable params: 12,785
Non-trainable params: 0
```

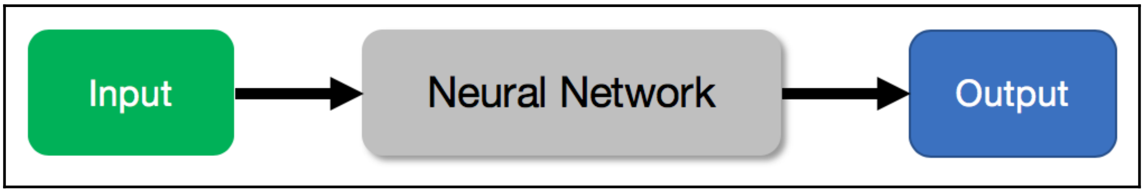

Noisy Images
Clean Images
Output Denoised Images

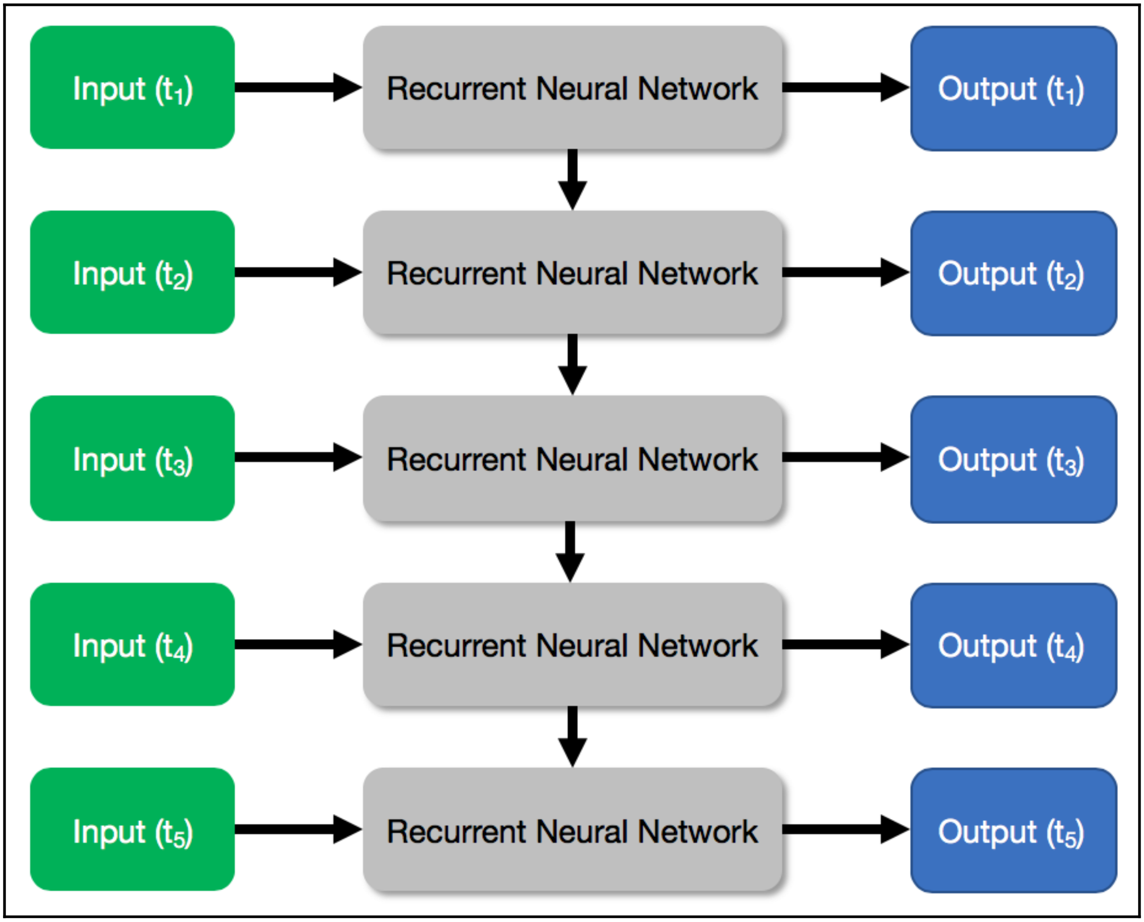
Chapter 6: Sentiment Analysis of Movie Reviews Using LSTM

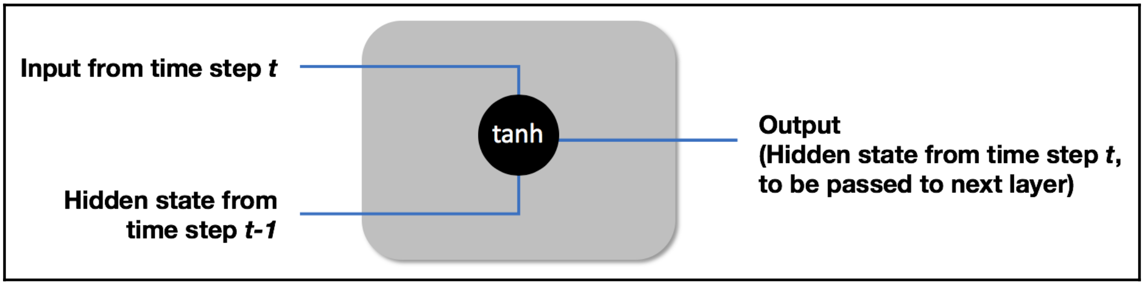


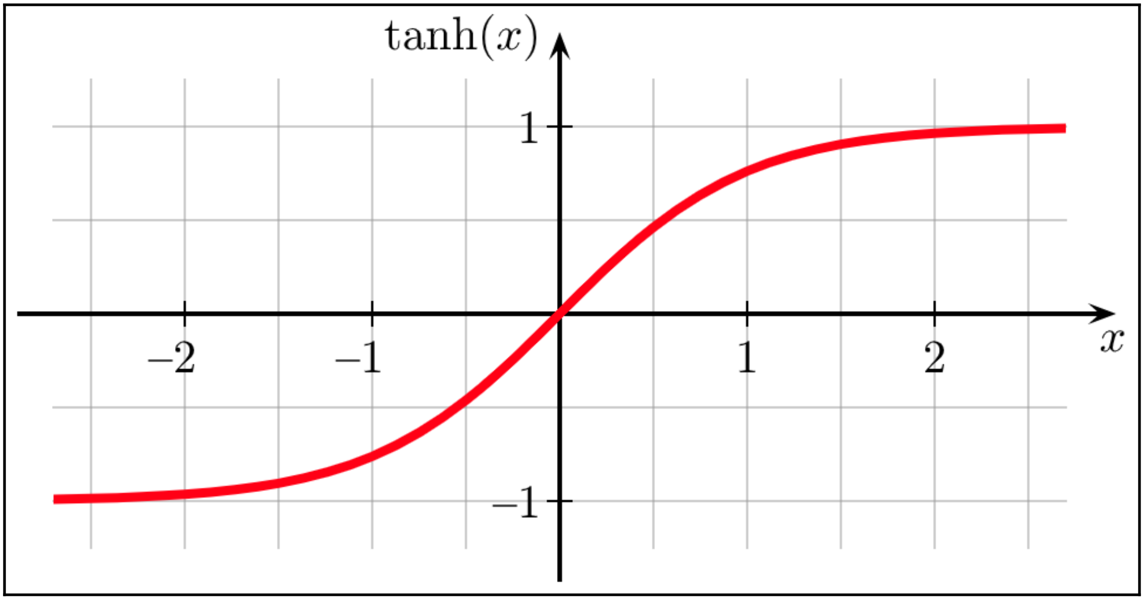

"THE BUILDING IS ON FIRE!"
"I AM ON FIRE TODAY!"

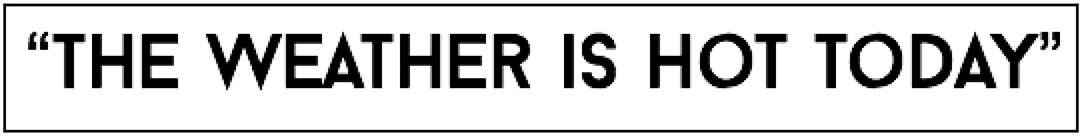

"THANKS FOR LOSING MY LUGGAGE! WHAT A WAY TO TREAT A LOYAL CUSTOMER"

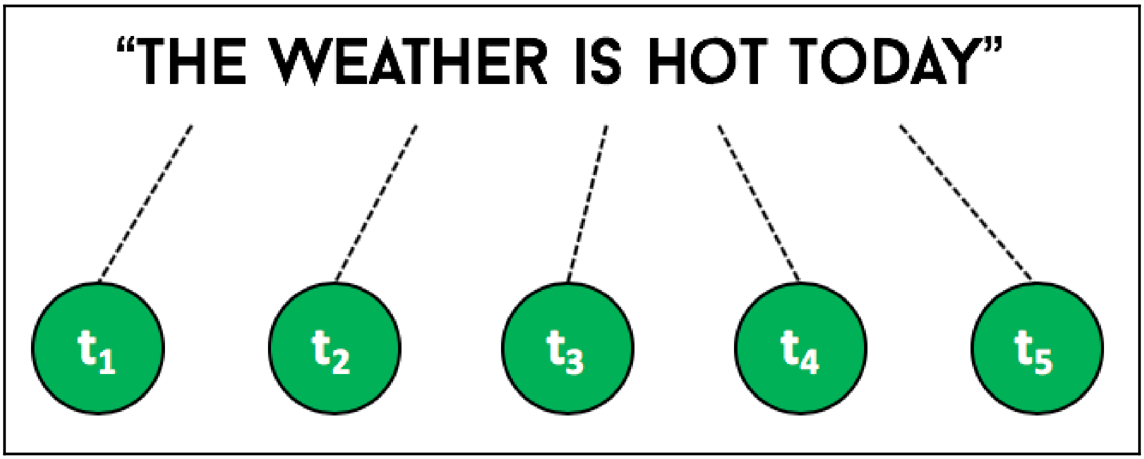

Input
Neural Network
Output

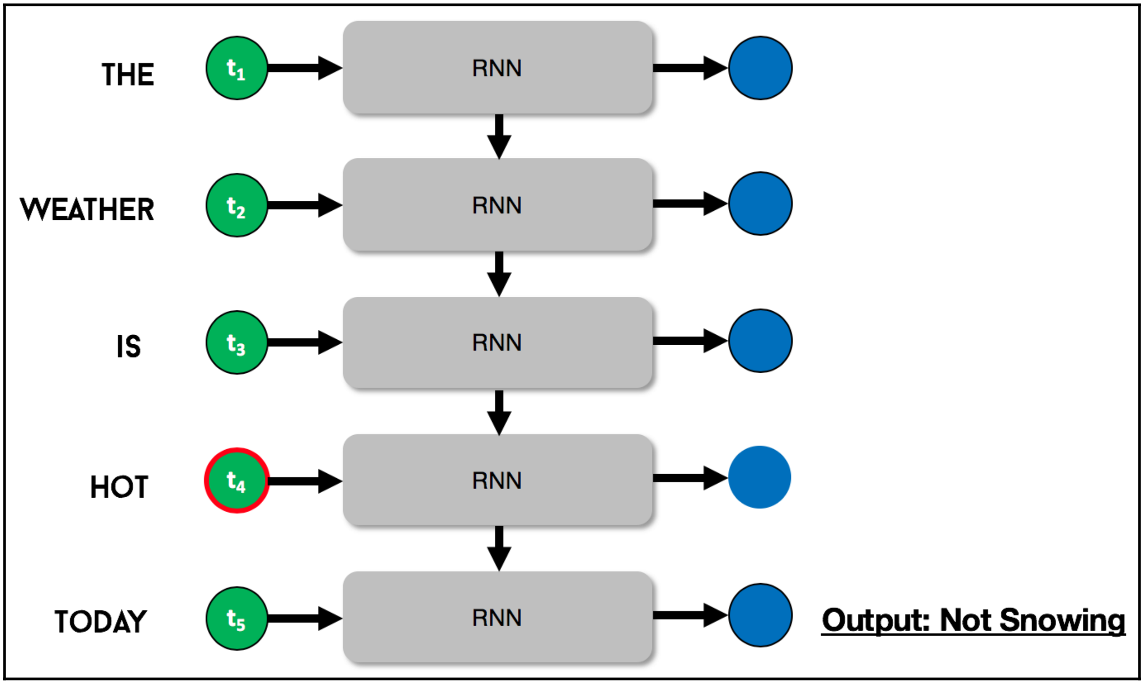

Input (t_1)
Recurrent Neural Network
Output (t_1)
Input (t_2)
Recurrent Neural Network
Output (t_2)
Input (t_3)
Recurrent Neural Network
Output (t_3)
Input (t_4)
Recurrent Neural Network
Output (t_4)
Input (t_5)
Recurrent Neural Network
Output (t_5)


Input from time step *t*
Hidden state from time step *t-1*
tanh
Output
(Hidden state from time step *t*, to be passed to next layer)

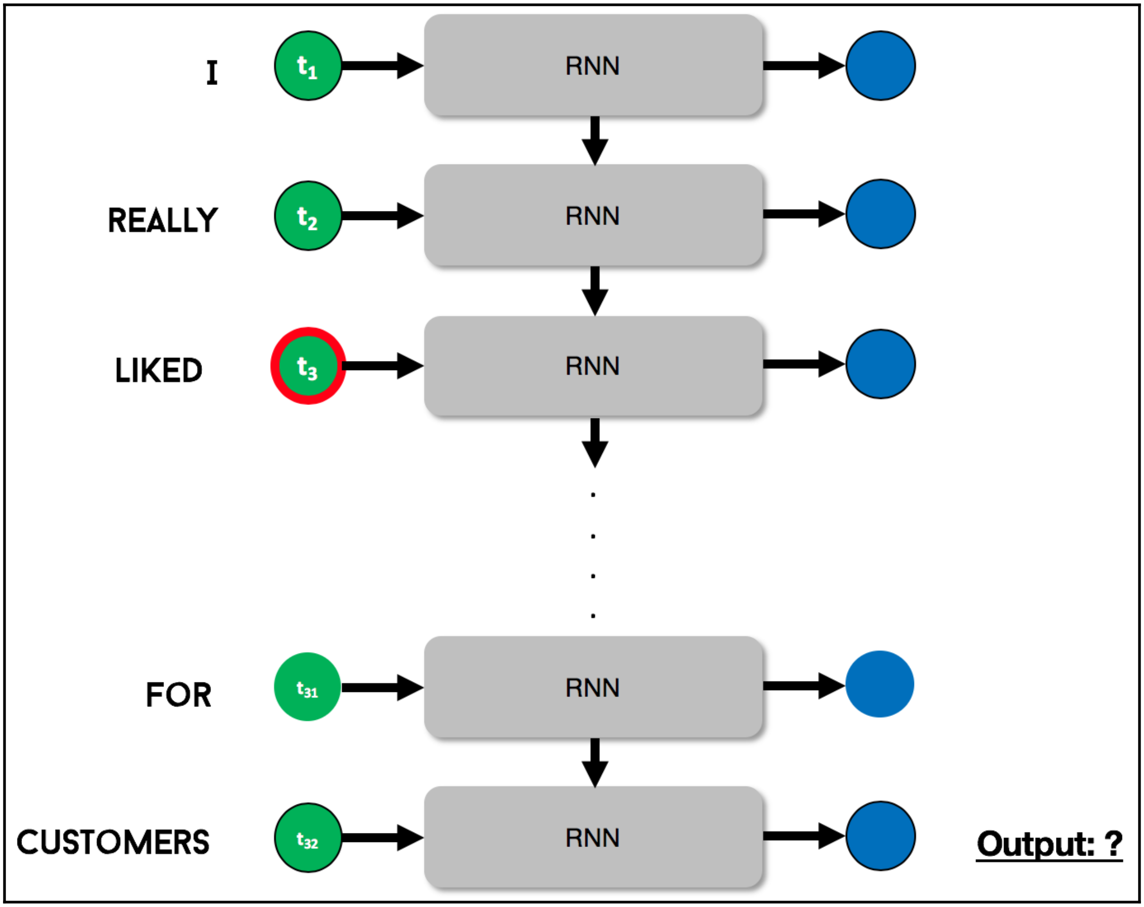

tanh(x)
1
−1
−2
−1
1
2
x

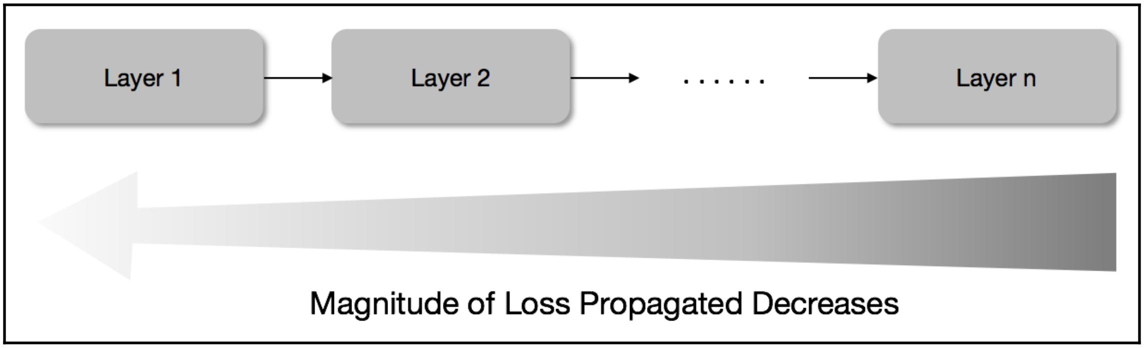

"THE WEATHER IS HOT TODAY"


"THE WEATHER IS HOT TODAY"
t1
t2
t3
t4
t5



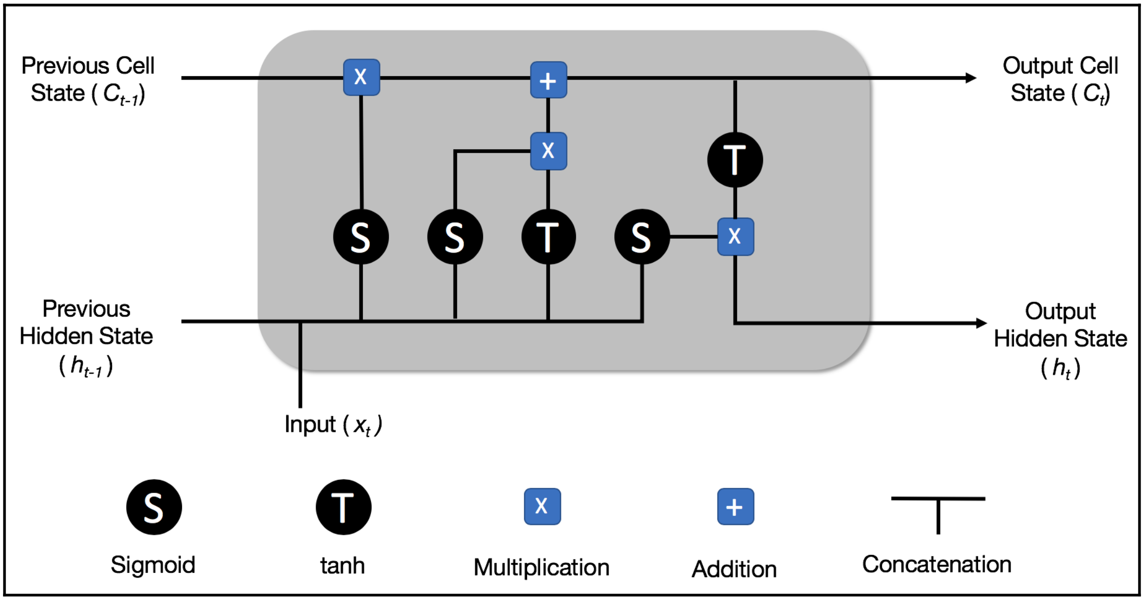
”I really liked the movie but I was disappointed in the service and cleanliness of the cinema. The cinema should be better maintained in order to provide a better experience for customers.”

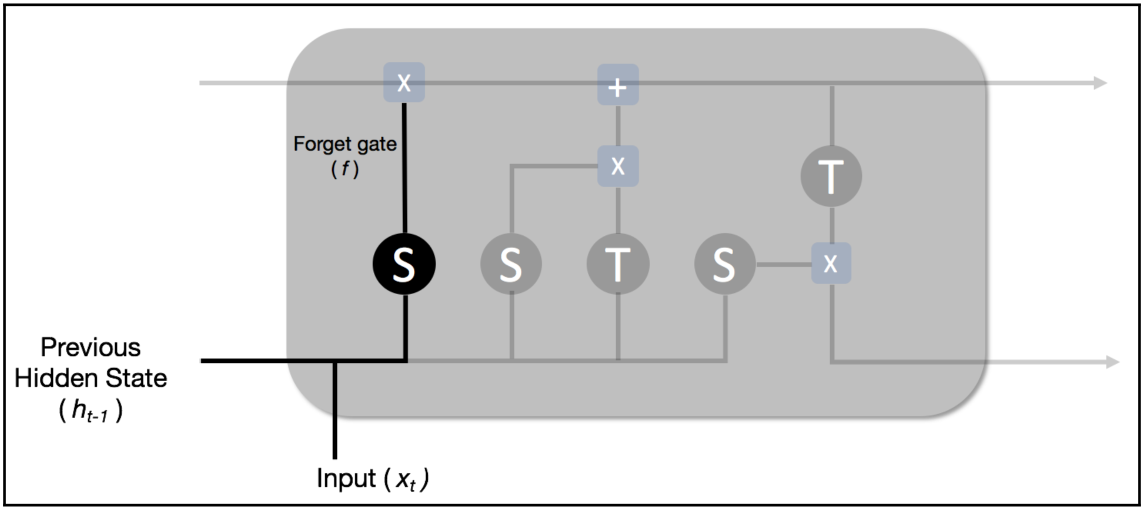

I
REALLY
LIKED
FOR
CUSTOMERS
t_1
t_2
t_3
t_{31}
t_{32}
RNN
RNN
RNN
RNN
RNN
Output: ?

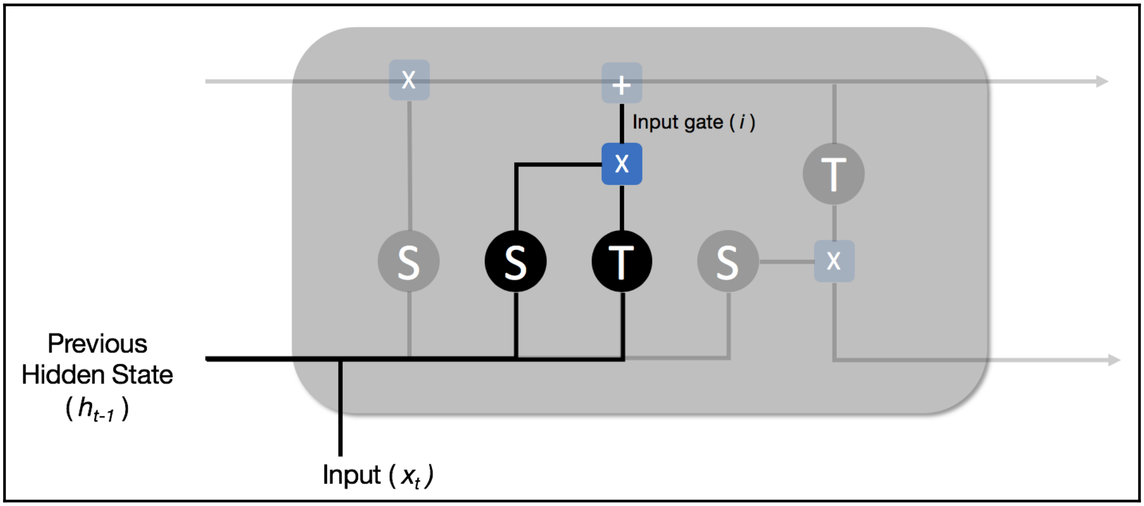

Layer 1
Layer 2
Layer n
Magnitude of Loss Propagated Decreases

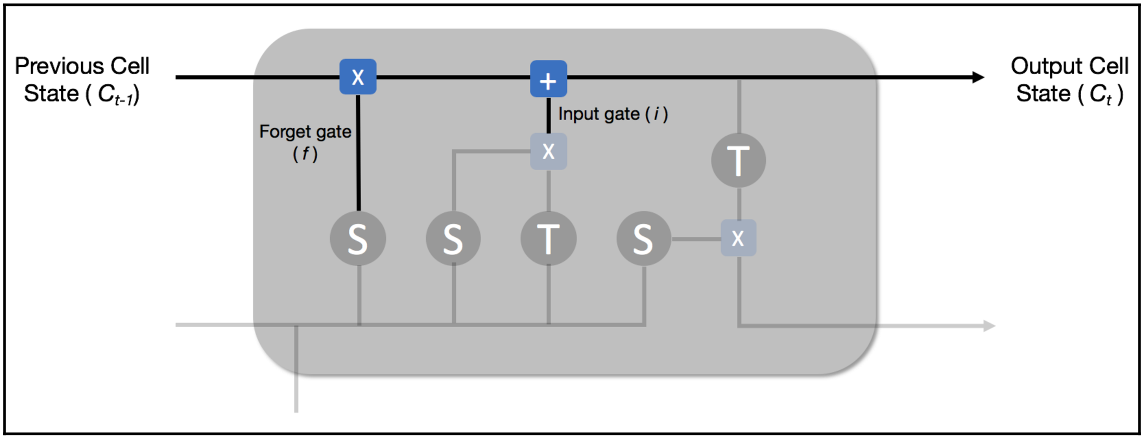
"I loved this movie! The action sequences were on point and the acting was terrific. Highly recommended!"

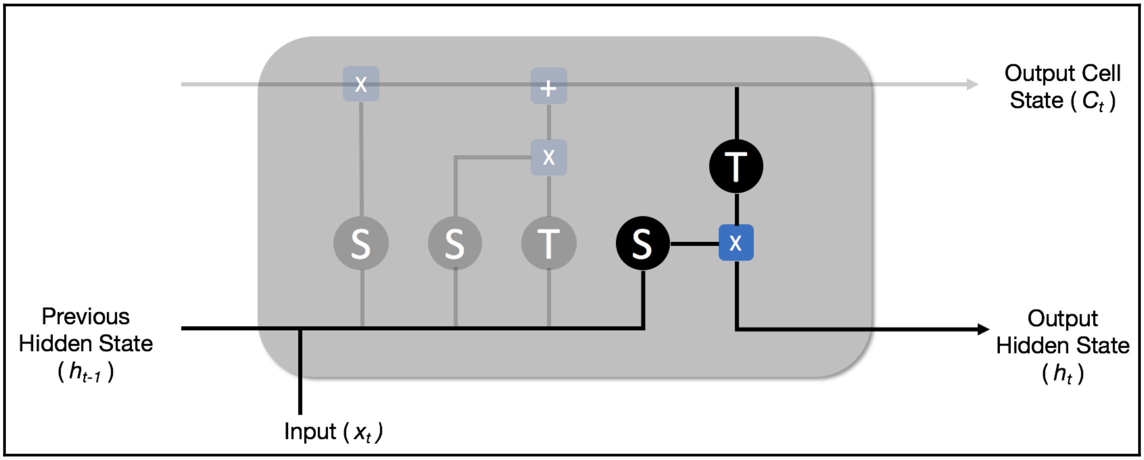
"I ==loved== this movie! The action sequences were ==on point== and the acting was ==terrific==. ==Highly recommended==!"




x
+
Forget gate
(f)
x
T
S
S
T
S
x
Previous
Hidden State
(h_{t-1})
Input (x_t)

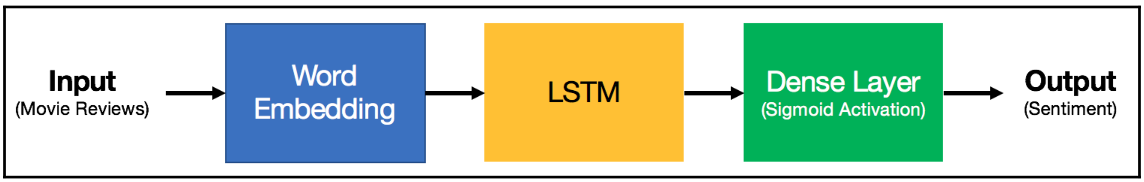

x
+
Input gate (i)
x
T
S
S
T
S
x
Previous
Hidden State
(h_{t-1})
Input (x_t)


Previous Cell State (C_{t-1})
Output Cell State (C_t)
Forget gate (f)
Input gate (i)
x
+
x
T
S
S
T
S
x

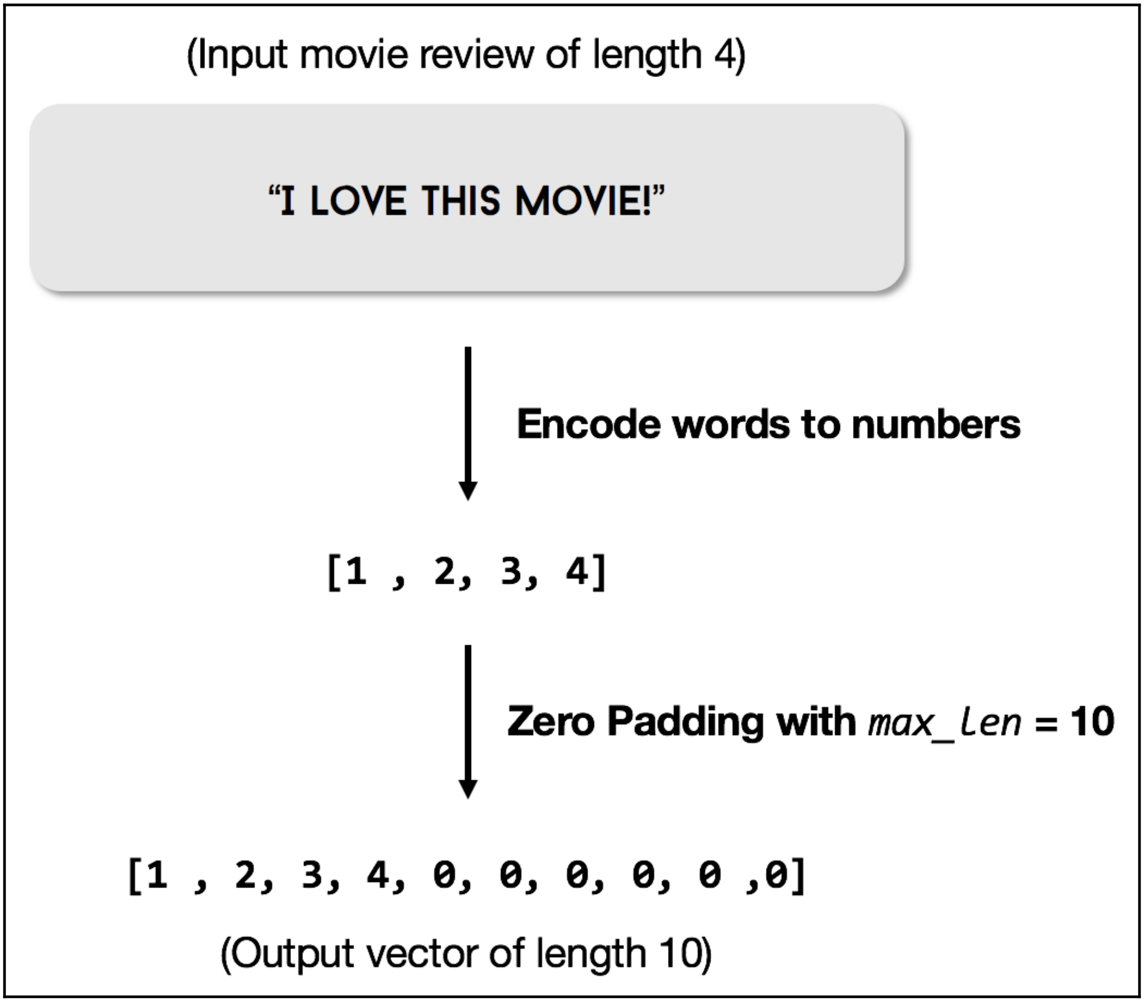

Output Cell State (C_t)
Previous Hidden State (h_{t-1})
Output Hidden State (h_t)
Input (x_t)
x
+
x
T
S
S
T
S
x


Happy
1
0
"Happy"
"Happy, excited"
"Excited"
0
1
Excited


"Elated"
"Happy"
"Excited"
"Sad"
"Disappointed"
"Angry"
"Furious"

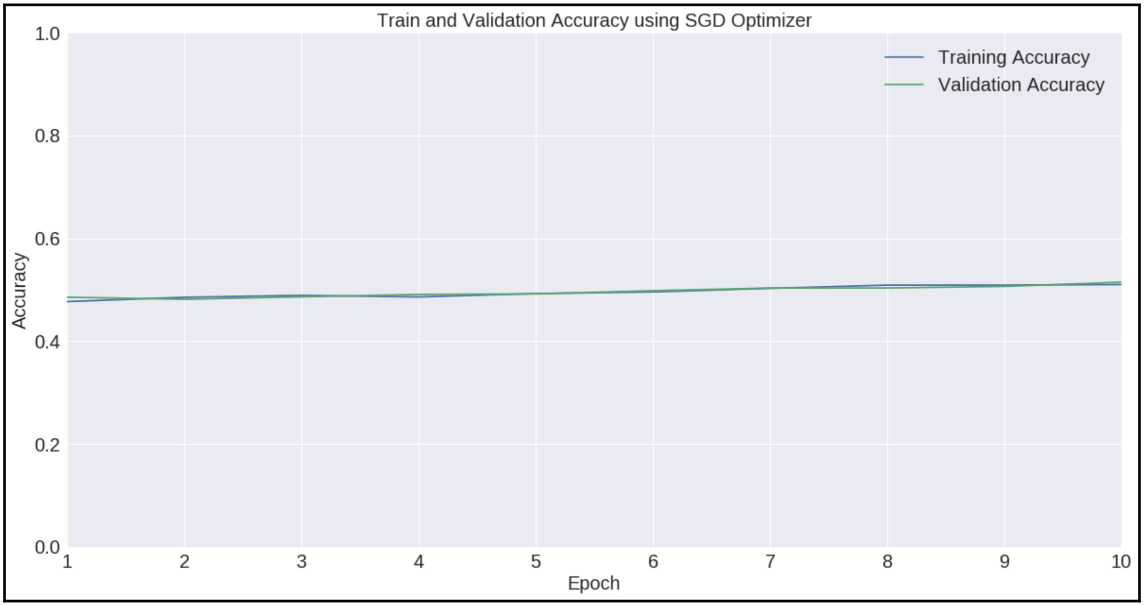

Input
(Movie Reviews)
Word Embedding
LSTM
Dense Layer
(Sigmoid Activation)
Output
(Sentiment)

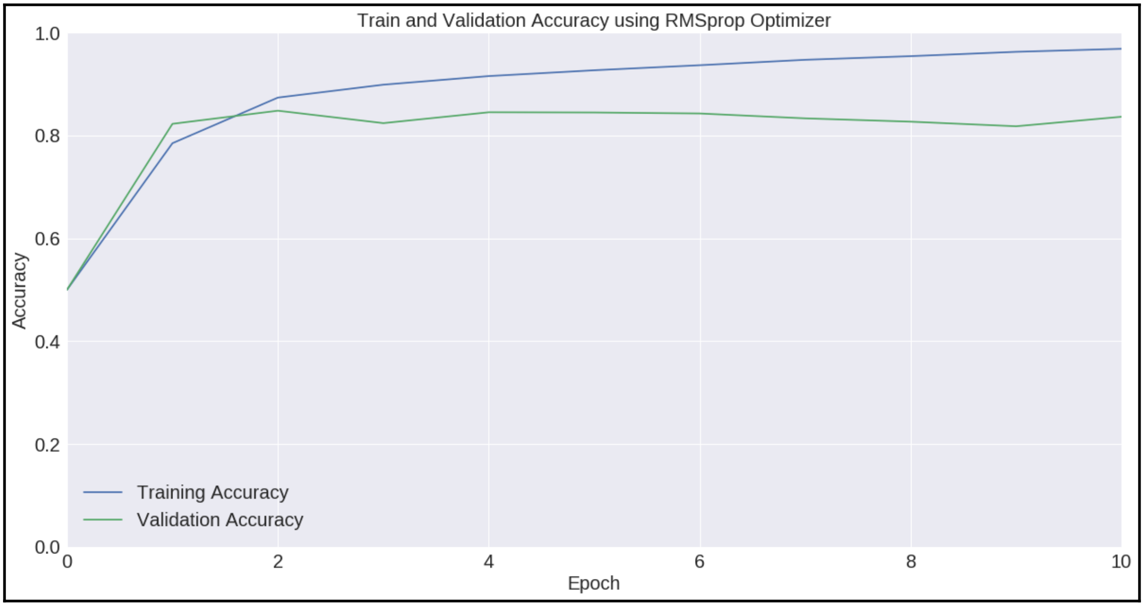
```
Number of training samples = 25000
Number of testing samples = 25000
```

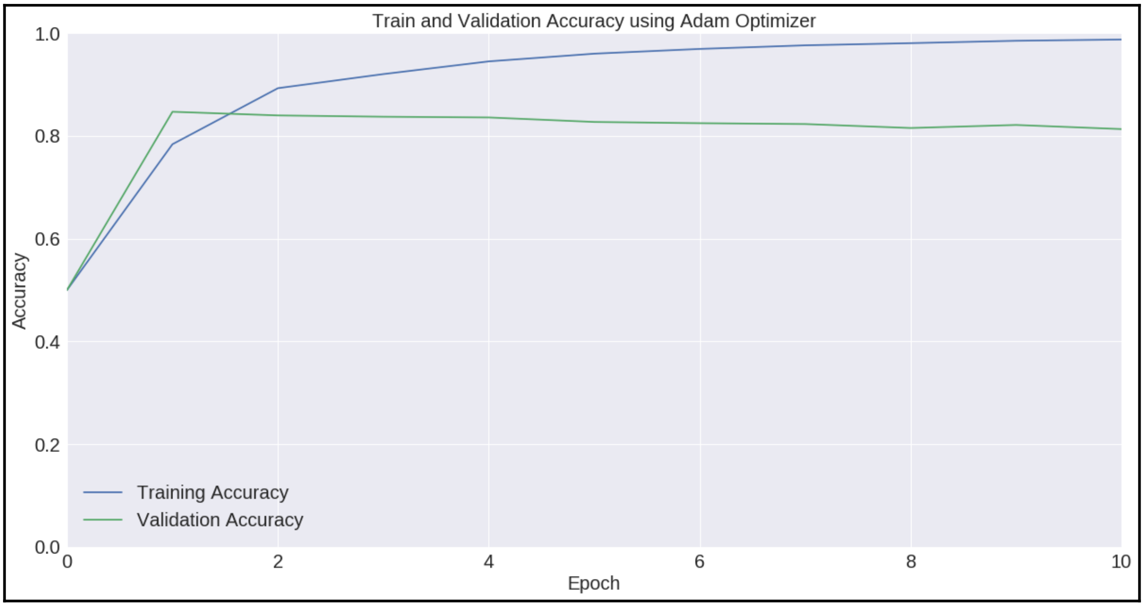


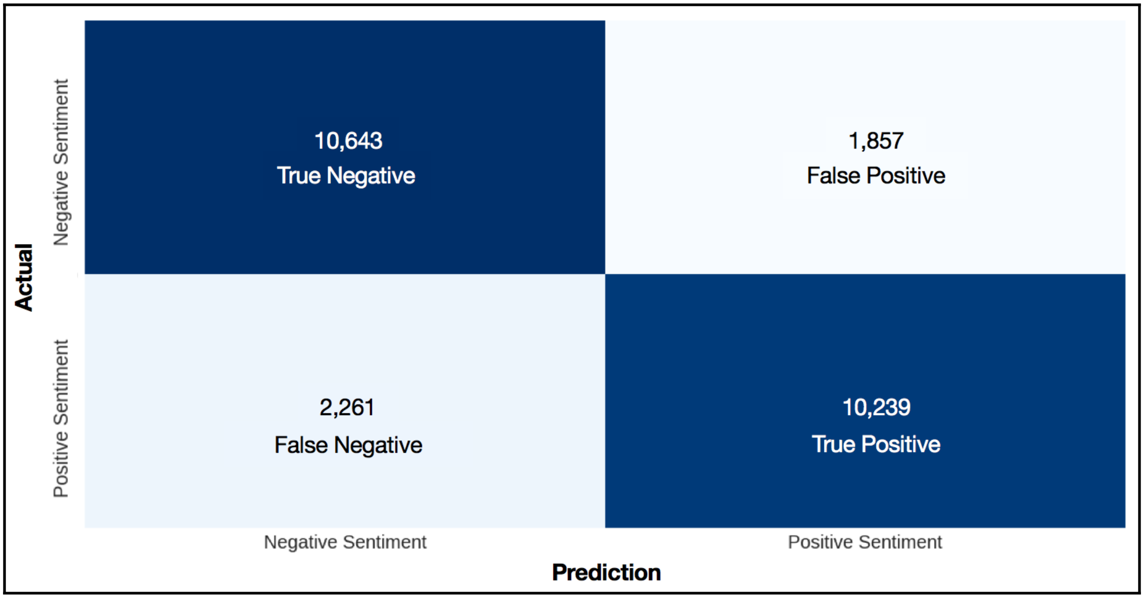

```
X_train vector shape = (25000, 100)
X_test vector shape = (25000, 100)
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_3 (Embedding)      (None, None, 128)         1280000
_________________________________________________________________
lstm_3 (LSTM)                (None, 128)               131584
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 129
=================================================================
Total params: 1,411,713
Trainable params: 1,411,713
Non-trainable params: 0
```

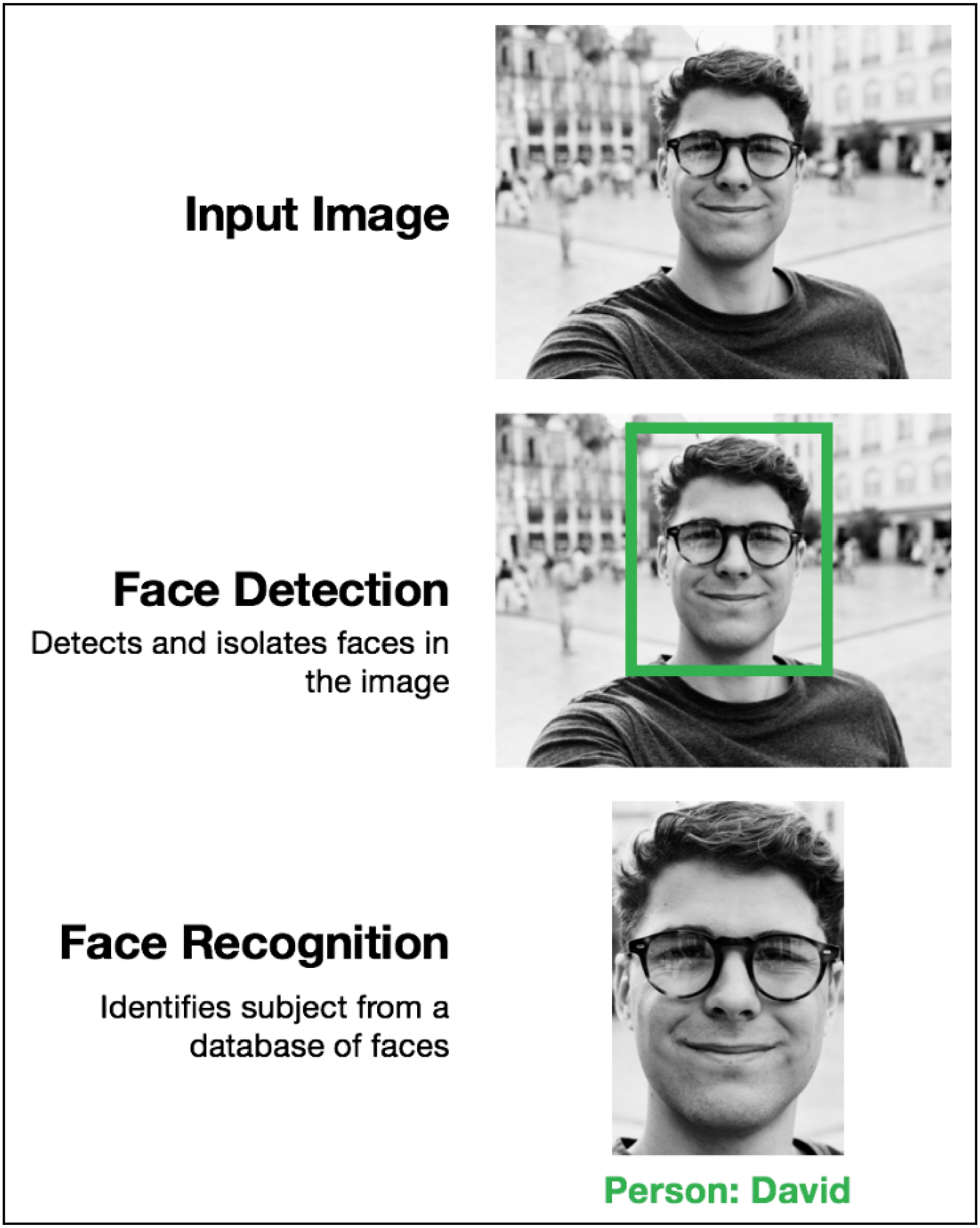


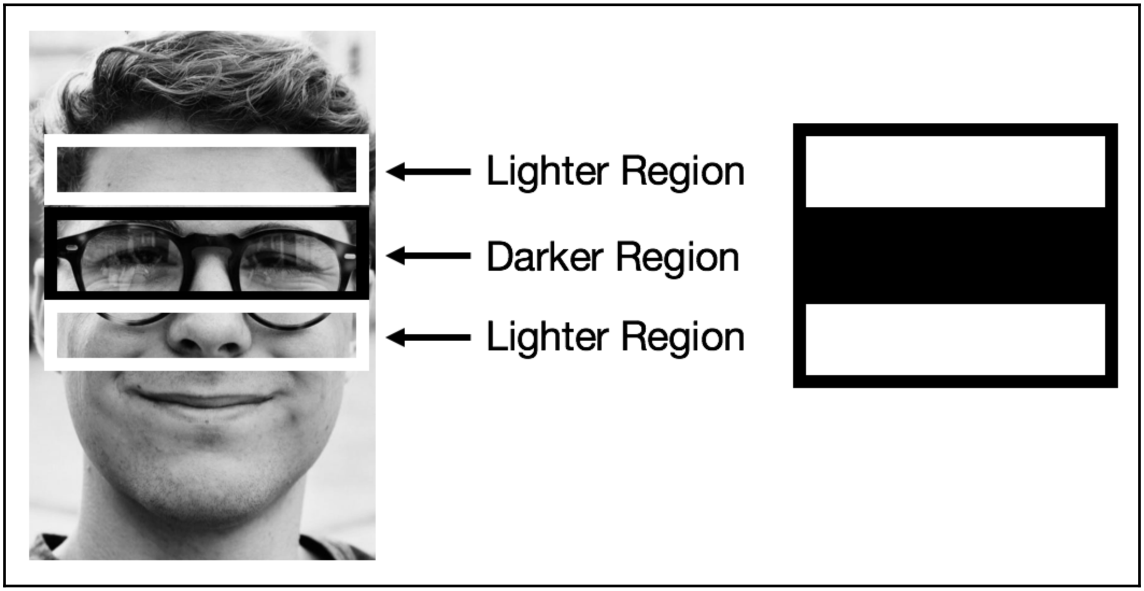

Train and Validation Accuracy using RMSprop Optimizer
Accuracy
Epoch
Training Accuracy
Validation Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10

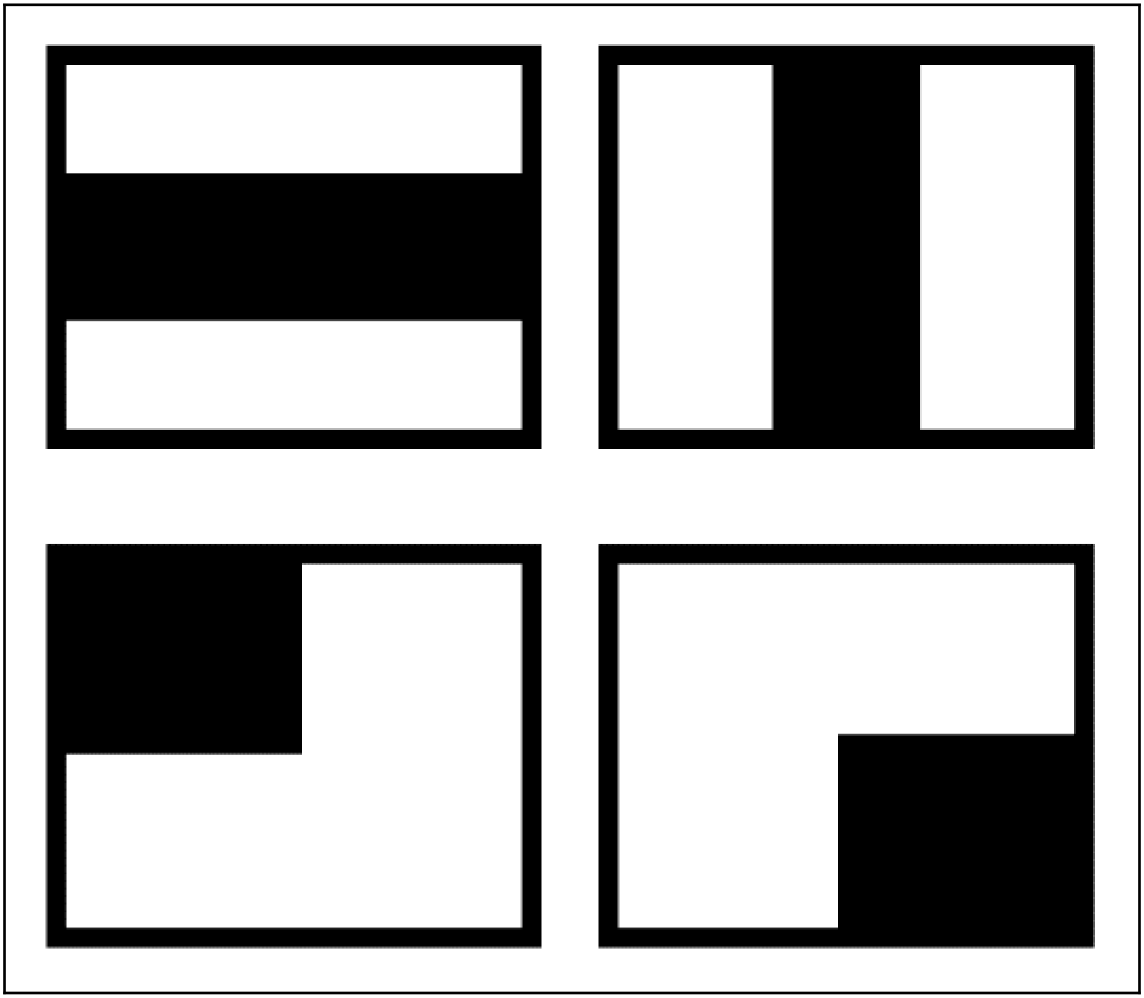

Train and Validation Accuracy using Adam Optimizer
Accuracy
Epoch
Training Accuracy
Validation Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10


Actual
Negative Sentiment
Positive Sentiment
10,643
True Negative
1,857
False Positive
2,261
False Negative
10,239
True Positive
Negative Sentiment
Positive Sentiment
Prediction

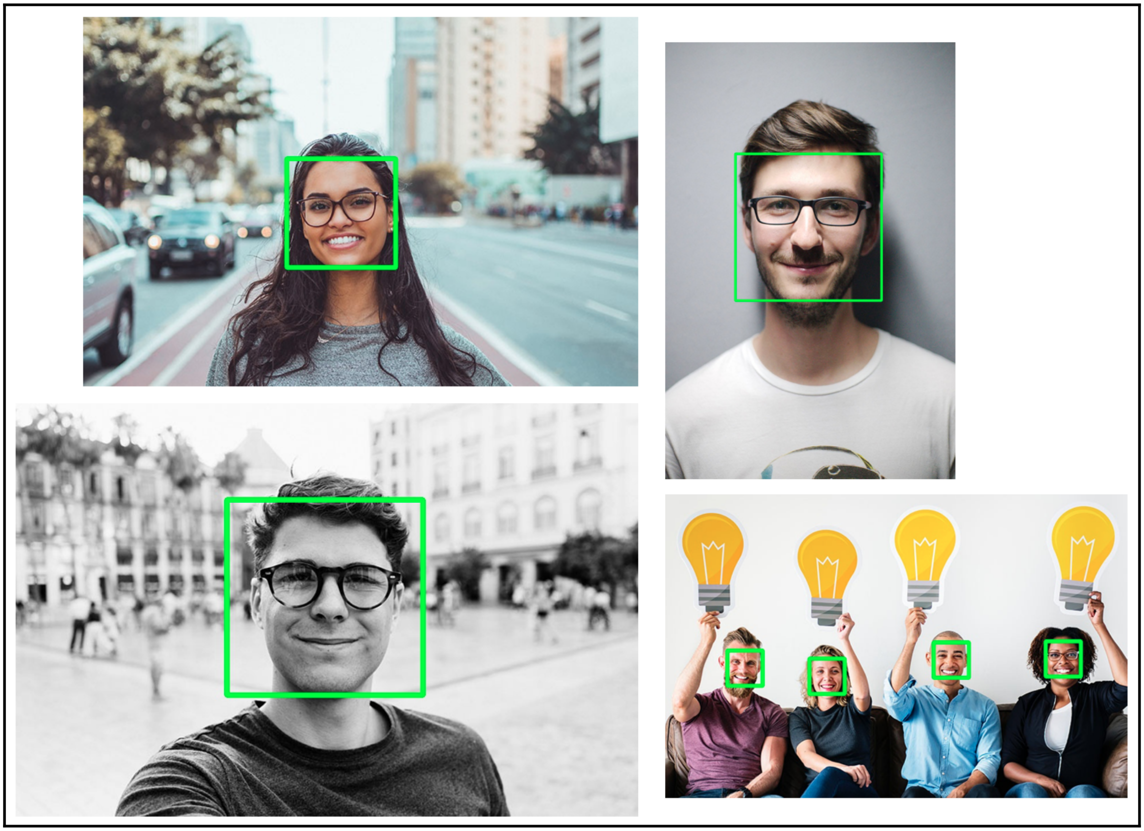
Chapter 7: Implementing a Facial Recognition System with Neural Networks

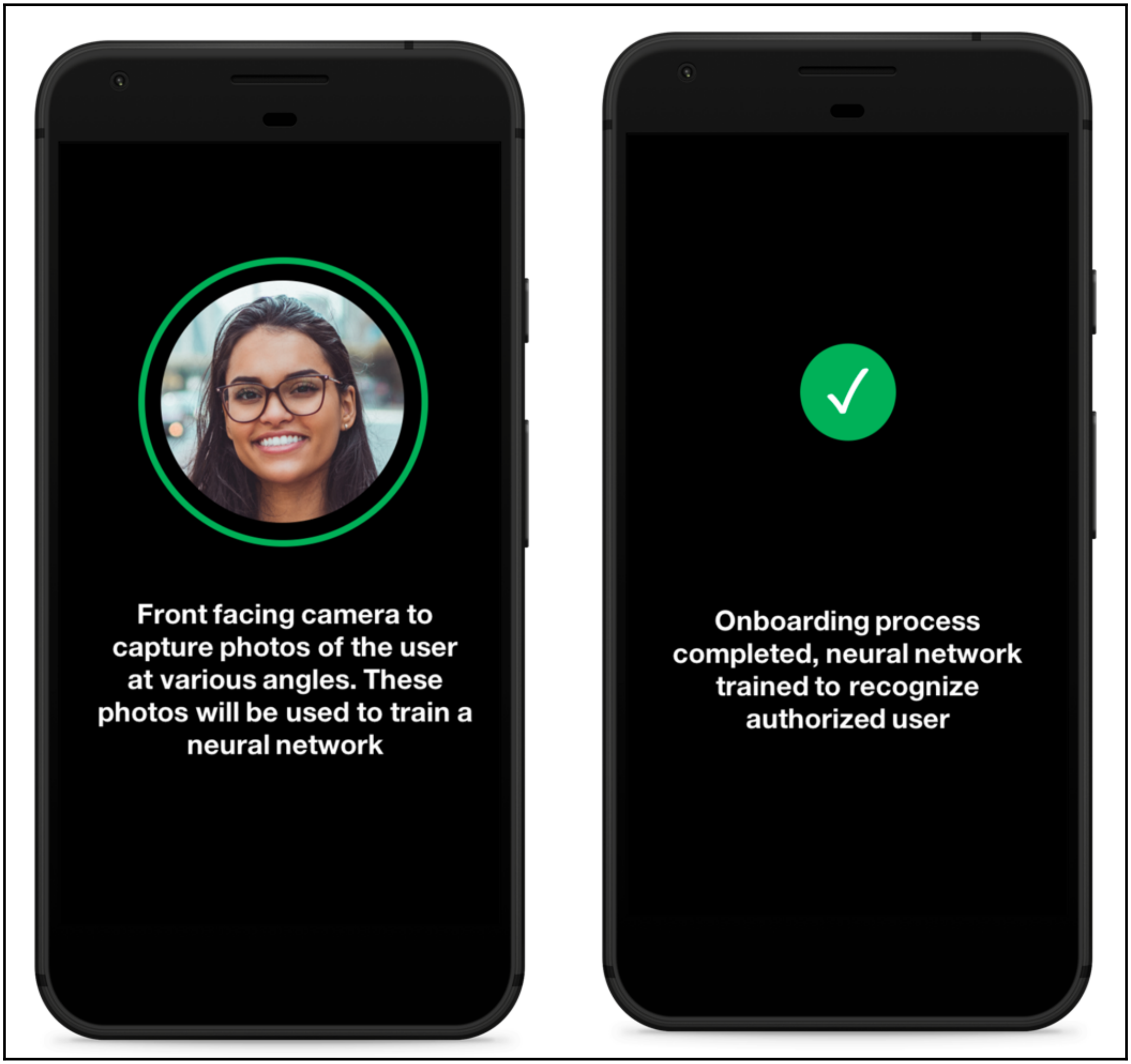

Input Image
Face Detection
Detects and isolates faces in the image
Face Recognition
Identifies subject from a database of faces
Person: David

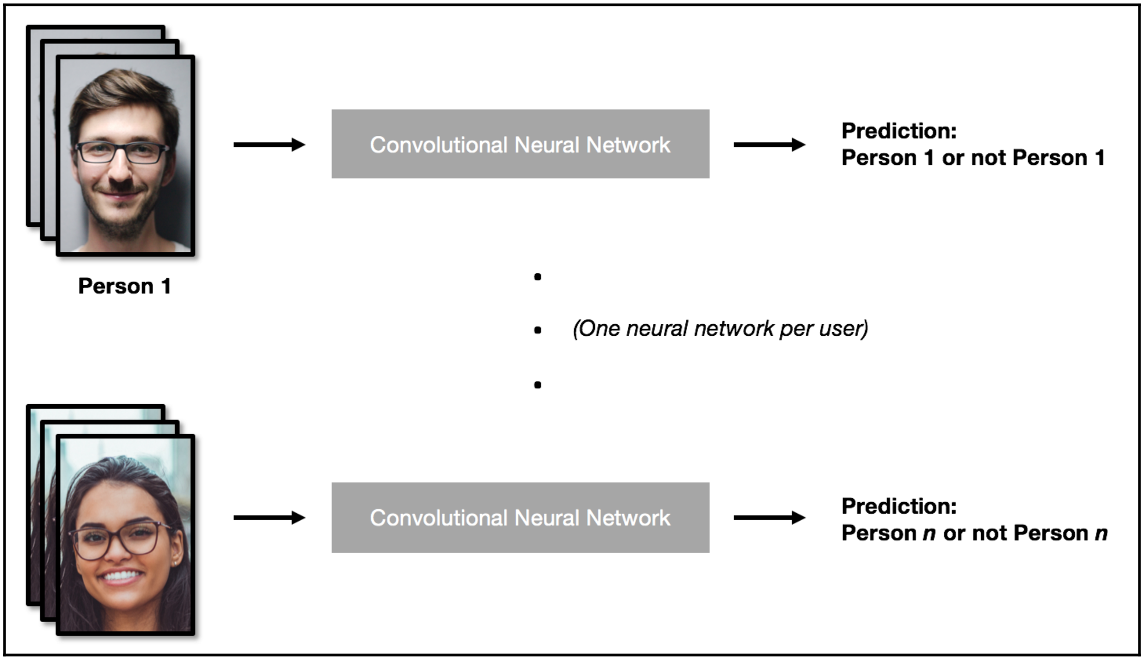

Lighter Region
Darker Region
Lighter Region

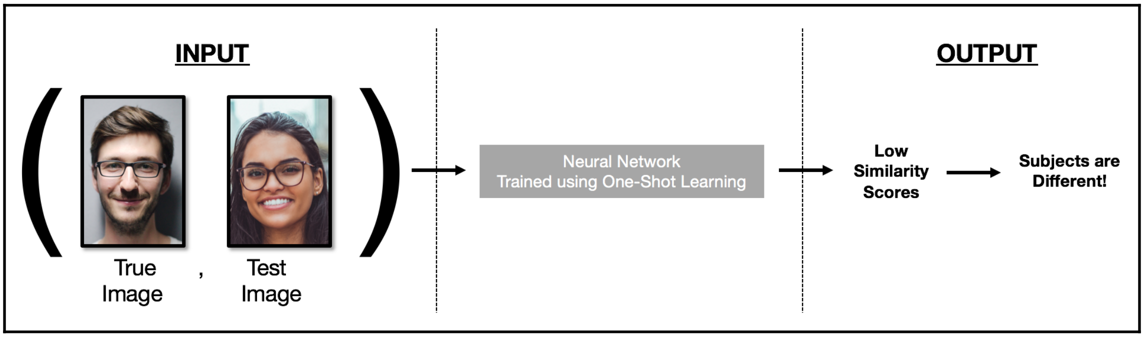


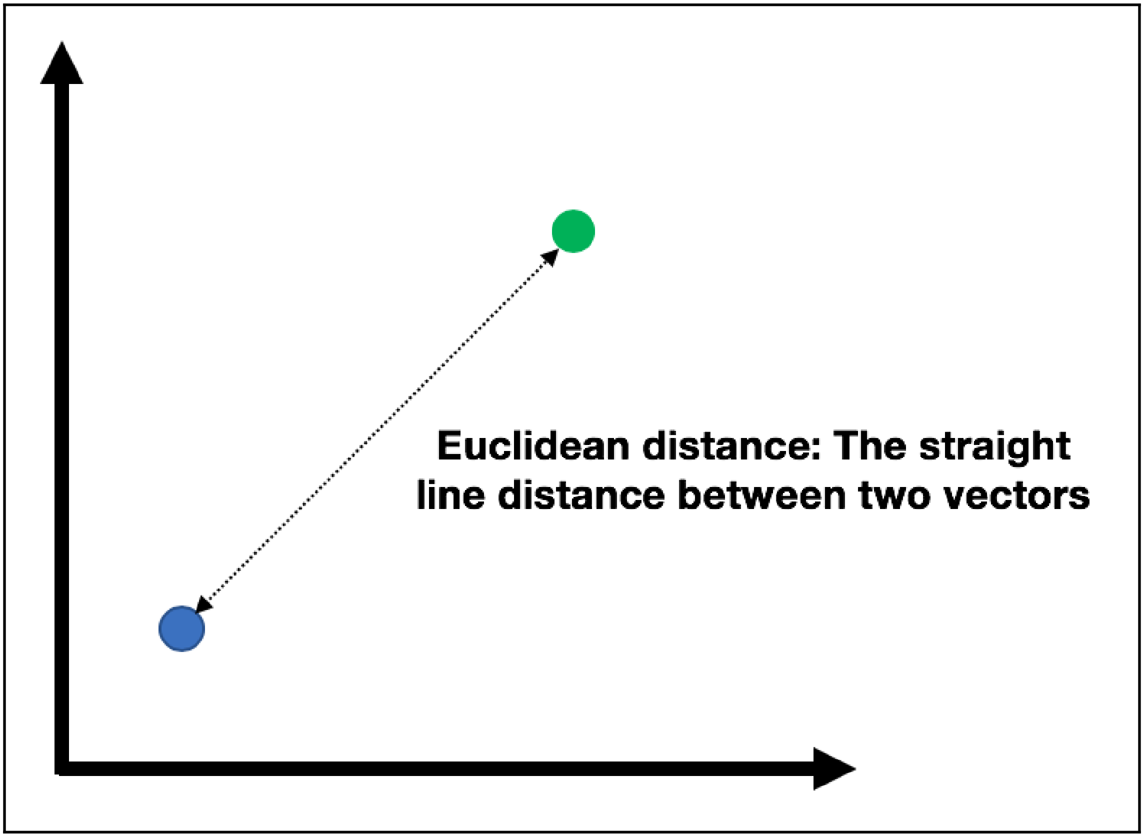


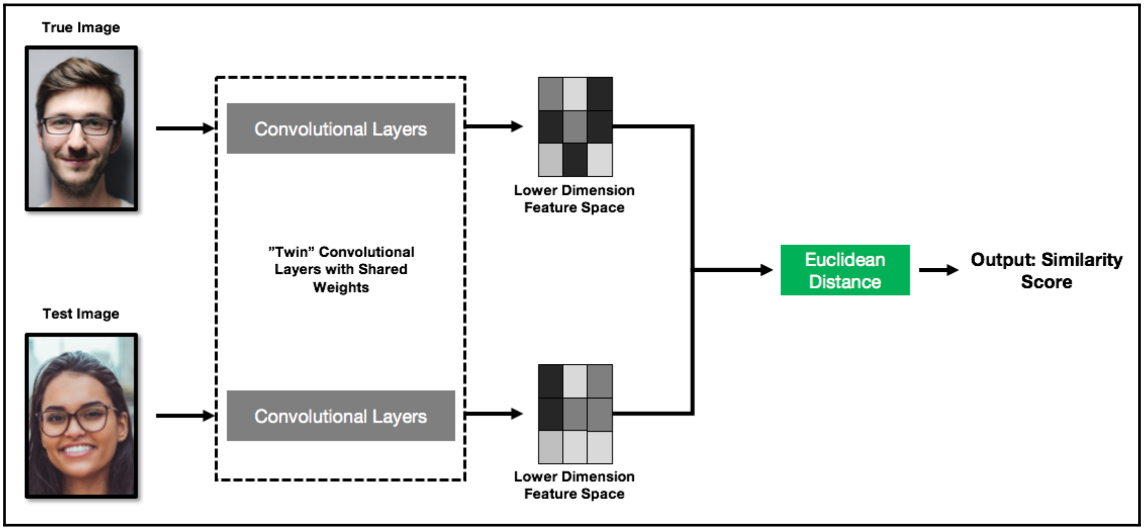



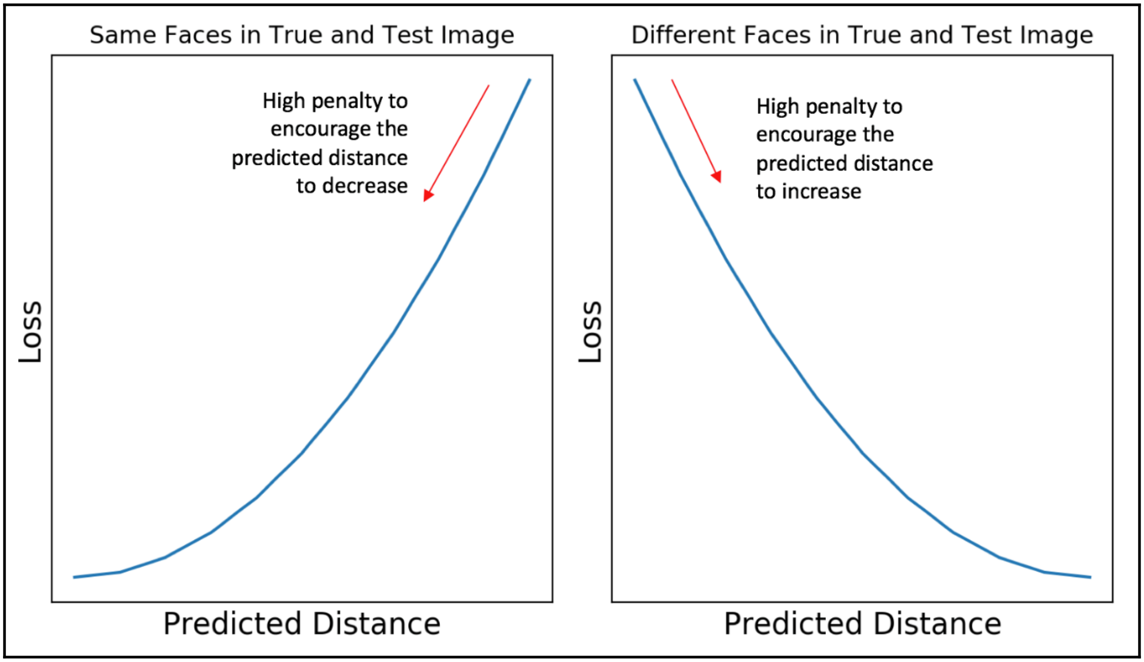

Front facing camera to capture photos of the user at various angles. These photos will be used to train a neural network
Onboarding process completed, neural network trained to recognize authorized user

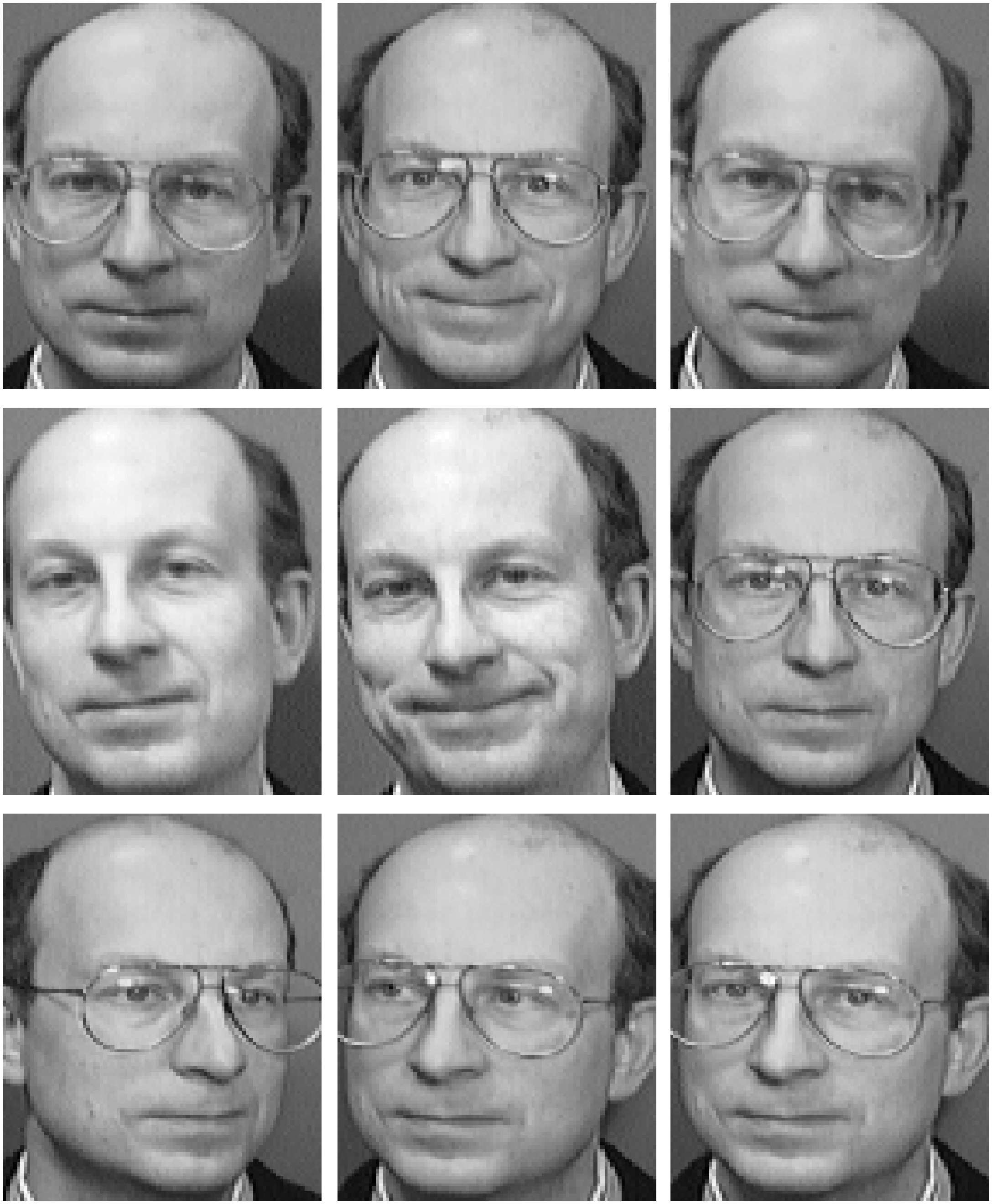

Convolutional Neural Network
Prediction:
Person 1 or not Person 1
Person 1
(One neural network per user)
Convolutional Neural Network
Prediction:
Person *n* or not Person *n*


INPUT
True Image
Test Image
Neural Network
Trained using One-Shot Learning
OUTPUT
Low Similarity Scores
Subjects are Different!

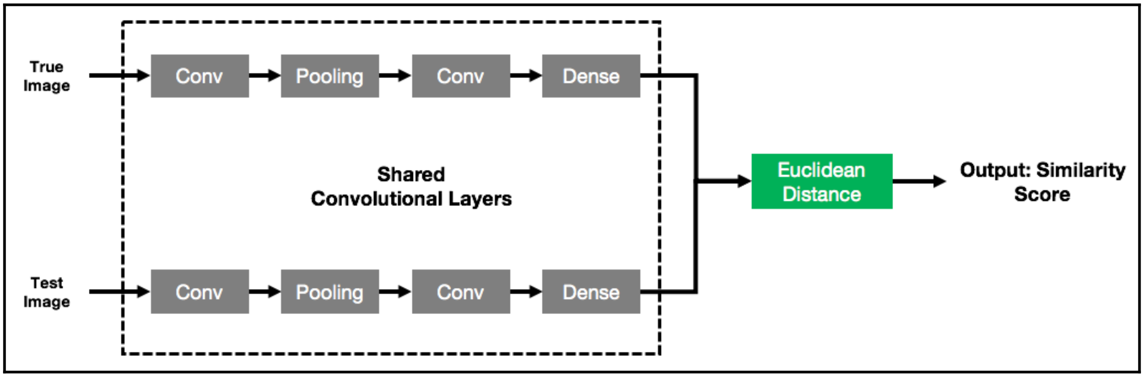

Euclidean distance: The straight line distance between two vectors

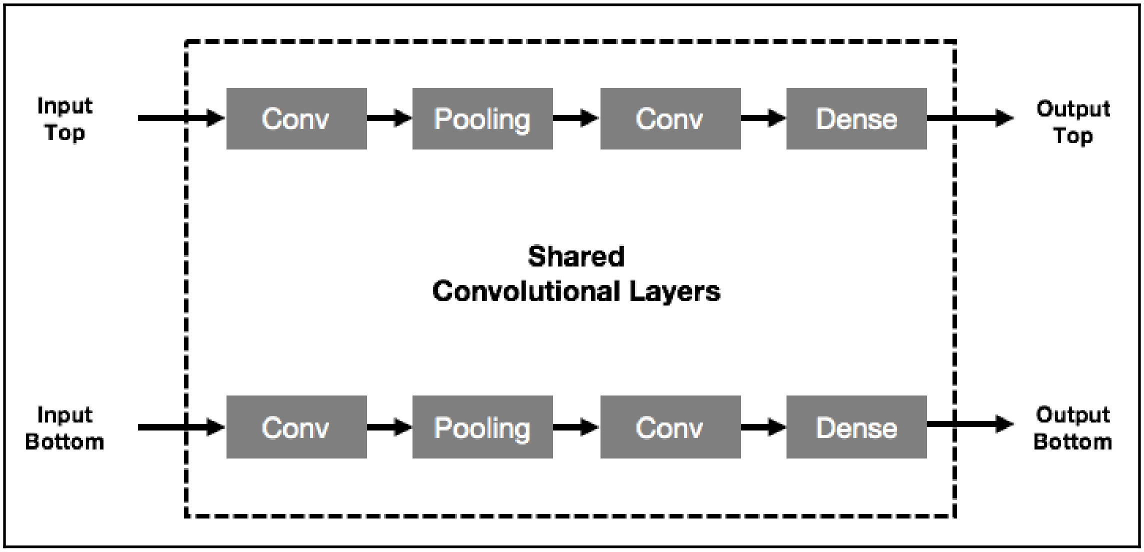

True Image
Convolutional Layers
Lower Dimension Feature Space
"Twin" Convolutional Layers with Shared Weights
Test Image
Convolutional Layers
Lower Dimension Feature Space
Euclidean Distance
Output: Similarity Score


Same Faces in True and Test Image
High penalty to encourage the predicted distance to decrease
Loss
Predicted Distance
Different Faces in True and Test Image
High penalty to encourage the predicted distance to increase
Loss
Predicted Distance

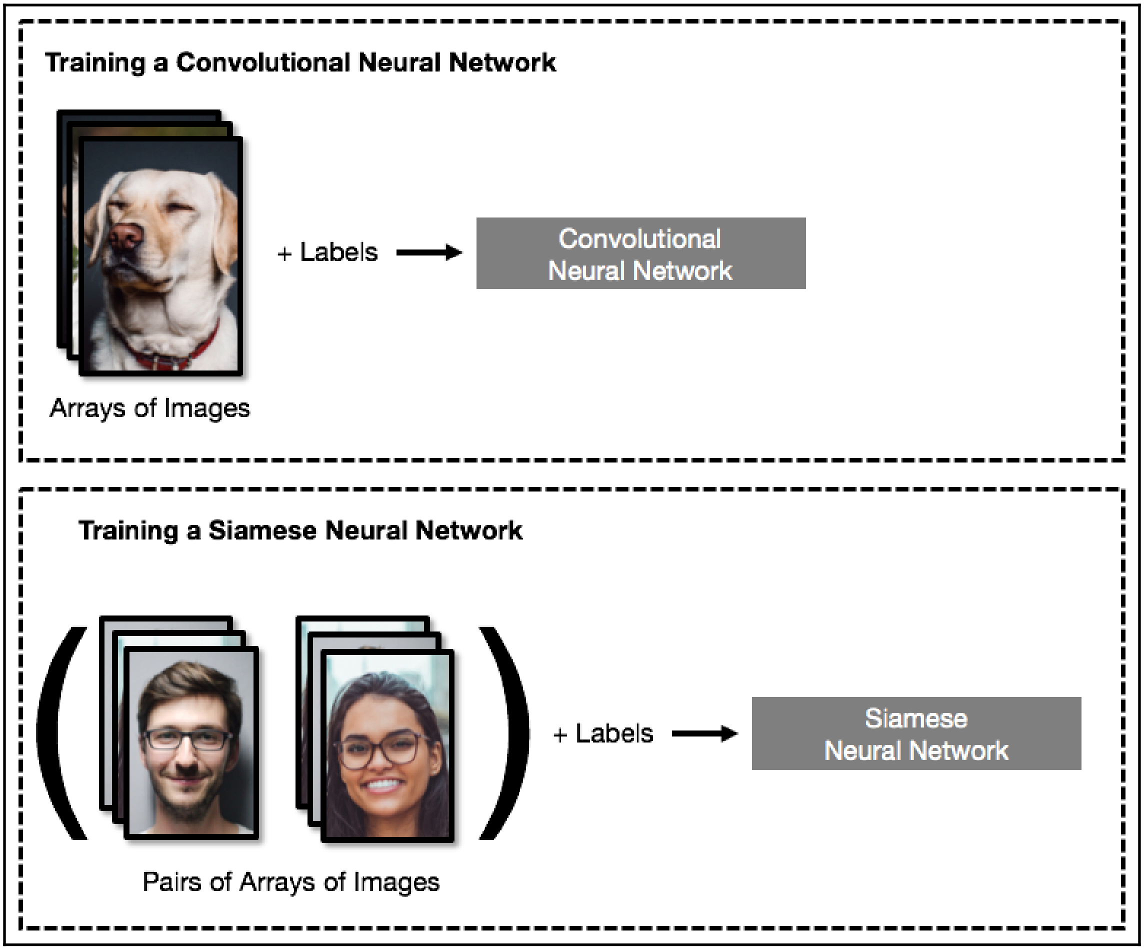



Subject 0
Subject 1
Subject 2
Subject 3
Subject 4
Subject 5
Subject 6
Subject 7
Subject 8

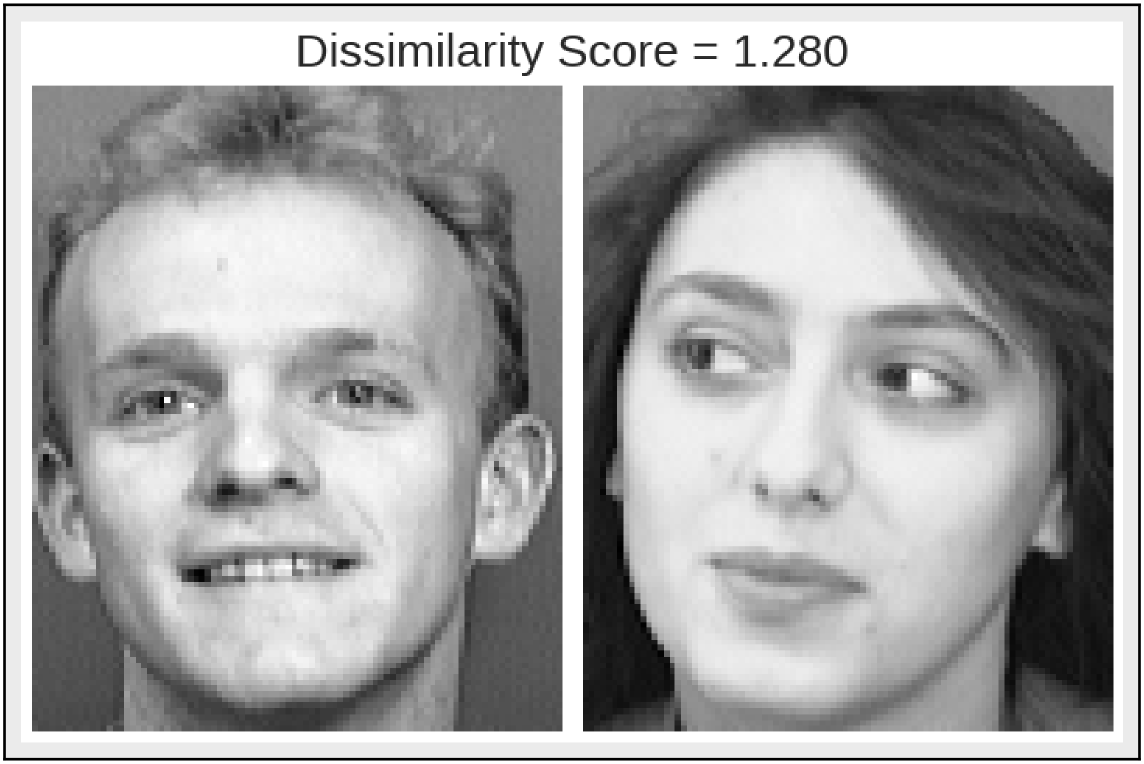

True Image
Test Image
Conv
Pooling
Conv
Dense
Conv
Pooling
Conv
Dense
Shared Convolutional Layers
Euclidean Distance
Output: Similarity Score

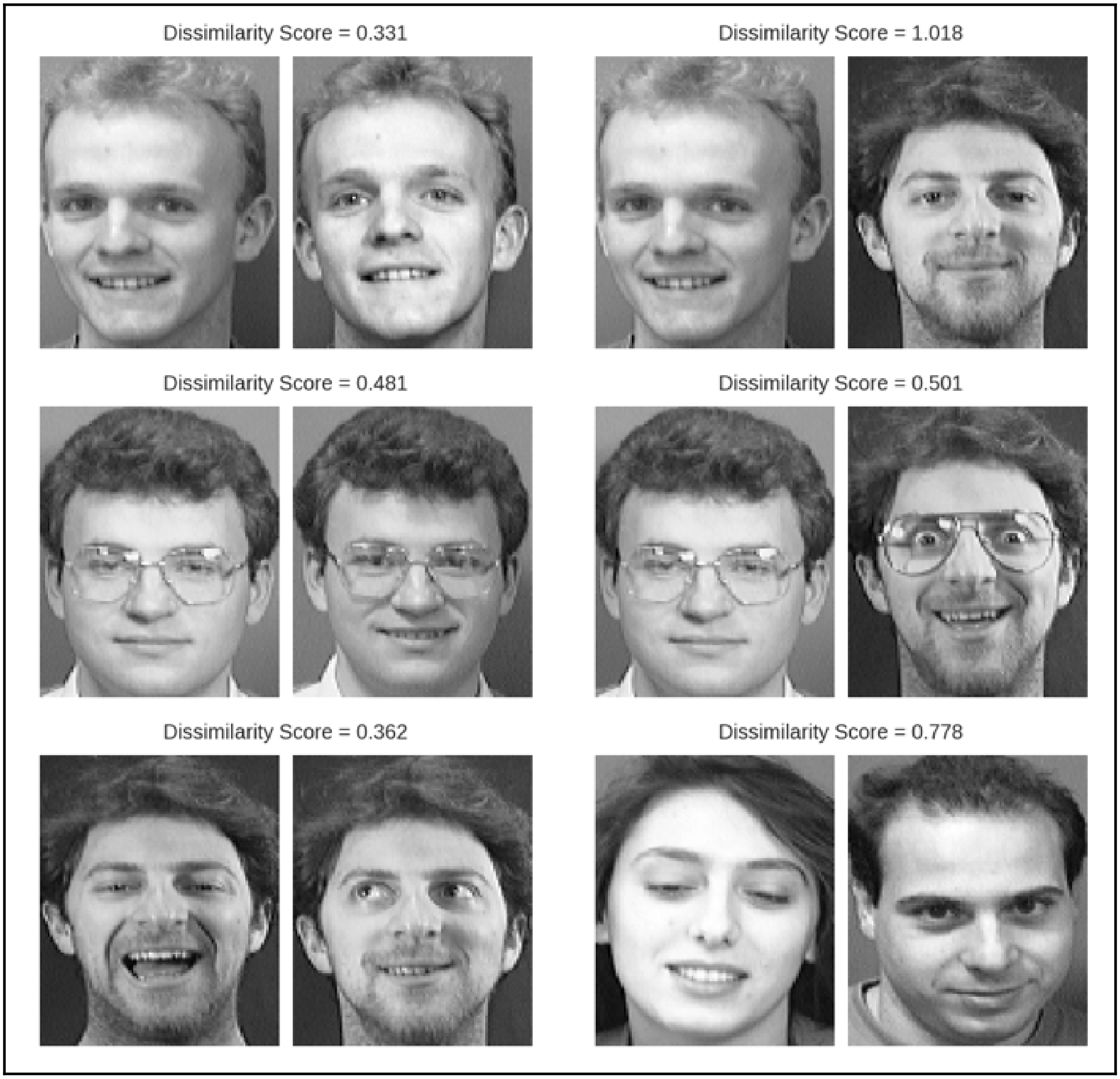

Input Top
Input Bottom
Conv
Pooling
Conv
Dense
Conv
Pooling
Conv
Dense
Shared Convolutional Layers
Output Top
Output Bottom

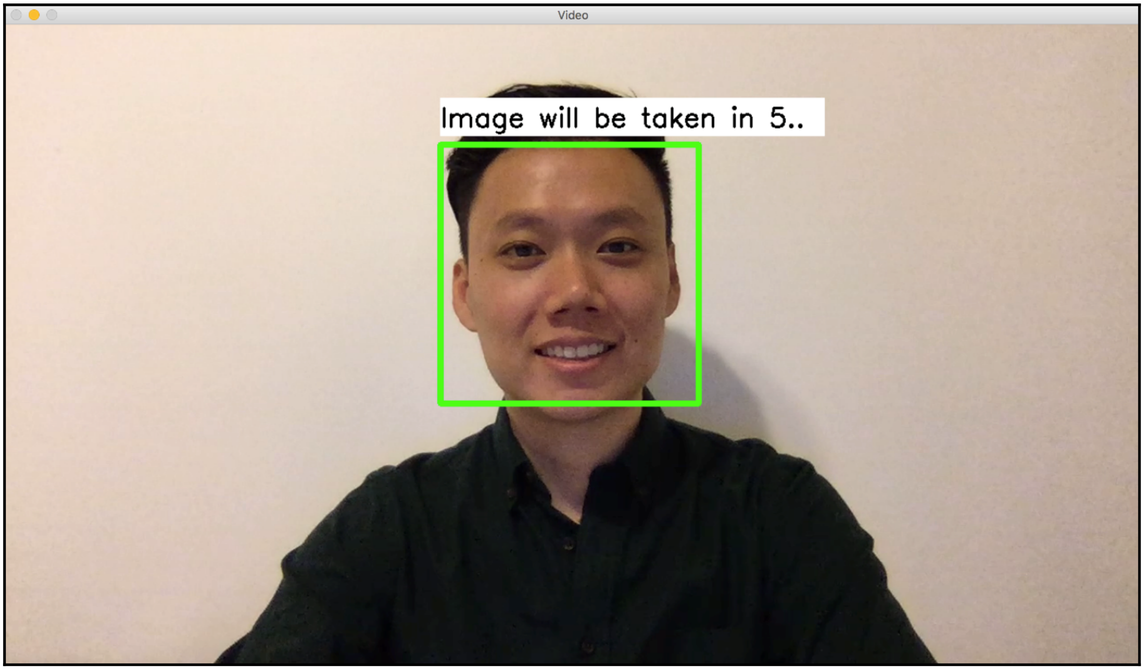
```
______________________________________________________________________________________________________
Layer (type)                       Output Shape          Param #       Connected to
======================================================================================================
input_1 (InputLayer)               (None, 112, 92, 1)    0
______________________________________________________________________________________________________
input_2 (InputLayer)               (None, 112, 92, 1)    0
______________________________________________________________________________________________________
Shared_Conv_Network (Sequential    (None, 128)           18707264      input_1[0][0]
                                                                       input_2[0][0]
______________________________________________________________________________________________________
Euclidean_Distance (Lambda)        (None, 1)             0             Shared_Conv_Network[1][0]
                                                                       Shared_Conv_Network[2][0]
======================================================================================================
Total params: 18,707,264
Trainable params: 18,707,264
Non-trainable params: 0
______________________________________________________________________________________________________
```

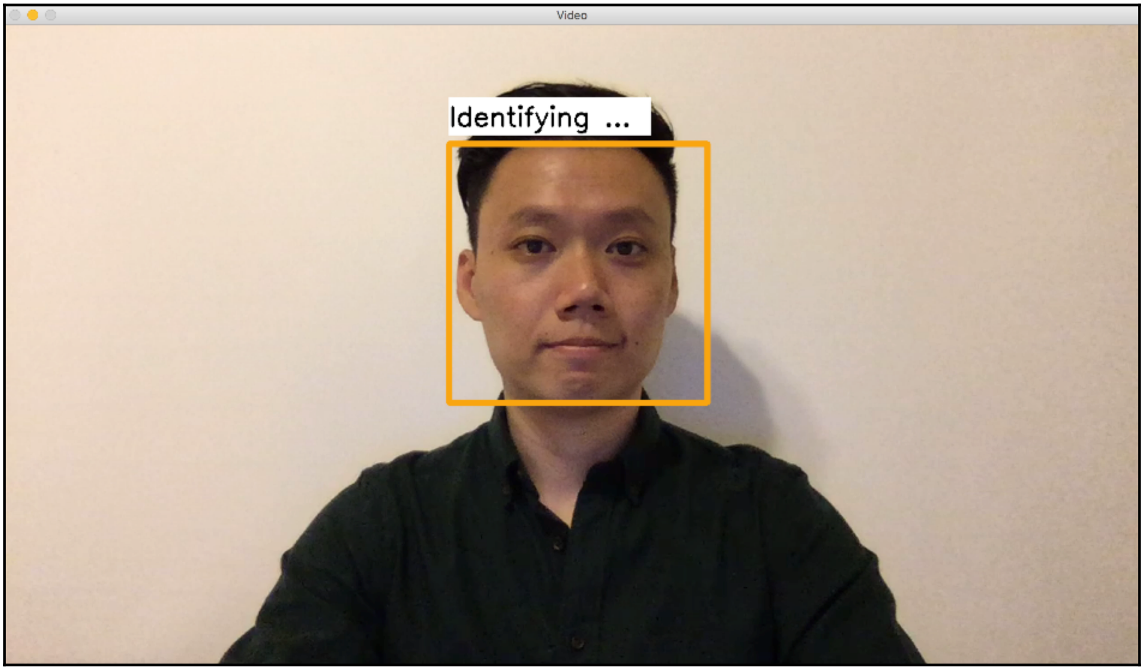


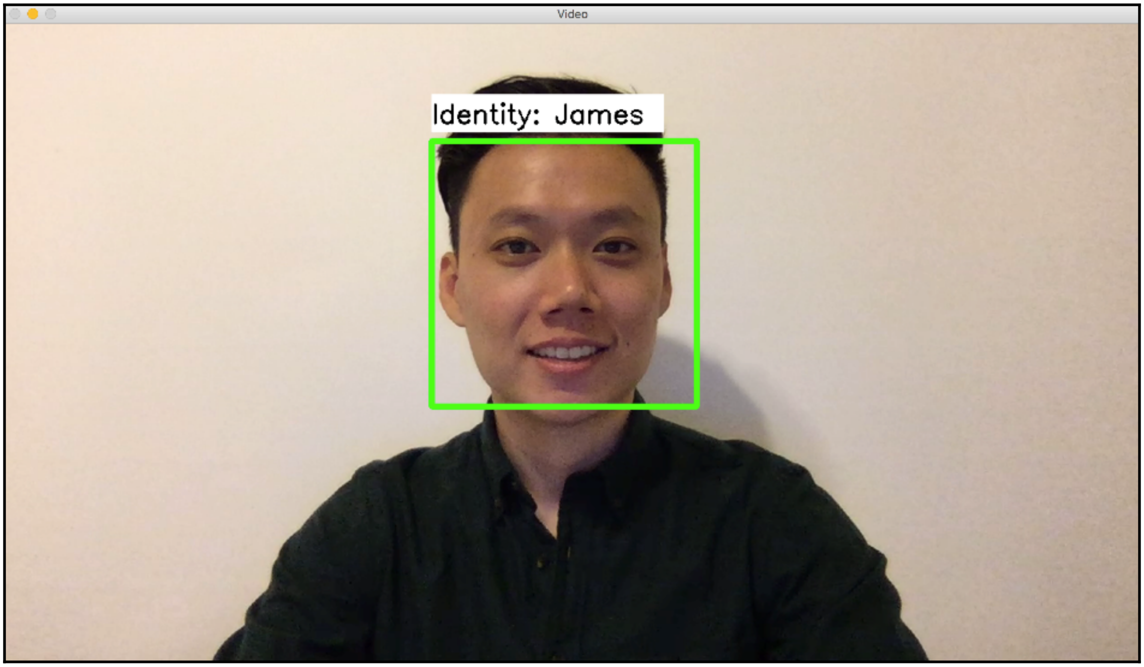
```
Epoch 1/10
630/630 [==============================] - 2s 3ms/step - loss: 0.2505 - accuracy: 0.7619
Epoch 2/10
630/630 [==============================] - 1s 2ms/step - loss: 0.1344 - accuracy: 0.8937
Epoch 3/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0932 - accuracy: 0.9413
Epoch 4/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0612 - accuracy: 0.9730
Epoch 5/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0404 - accuracy: 0.9921
Epoch 6/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0283 - accuracy: 0.9984
Epoch 7/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0208 - accuracy: 1.0000
Epoch 8/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0155 - accuracy: 1.0000
Epoch 9/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0123 - accuracy: 1.0000
Epoch 10/10
630/630 [==============================] - 1s 2ms/step - loss: 0.0091 - accuracy: 1.0000
```


Dissimilarity Score = 1.280


Dissimilarity Score = 0.331
Dissimilarity Score = 1.018
Dissimilarity Score = 0.481
Dissimilarity Score = 0.501
Dissimilarity Score = 0.362
Dissimilarity Score = 0.778


Video
Image will be taken in 5..


Video
Identifying ...


Video
Identity: James