Deep Learning

M Octaviano Pratama, S.Kom., M.Kom

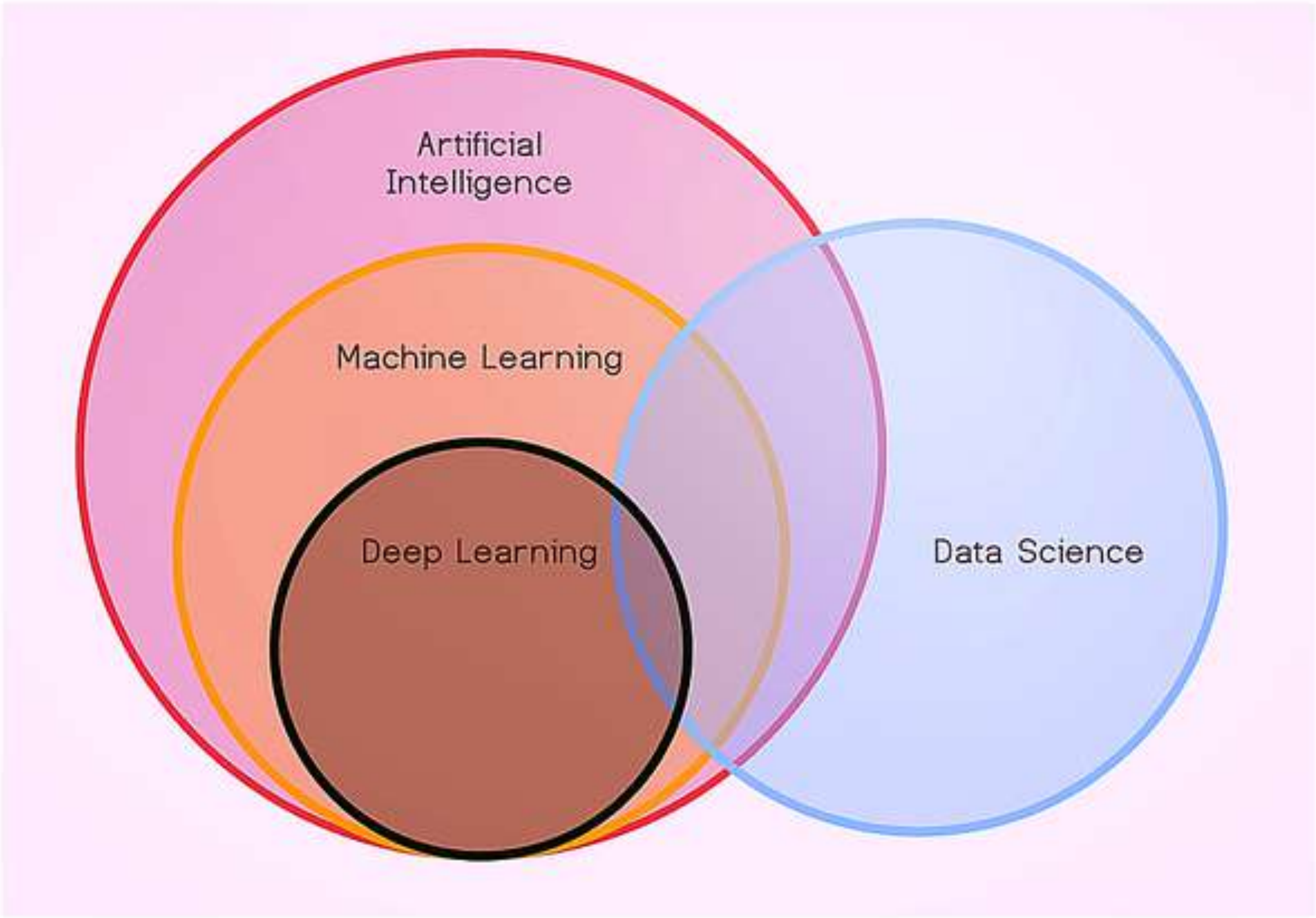
President Director PT Bisa Artifisial Indonesia

Institut Teknologi Telkom Purwokerto

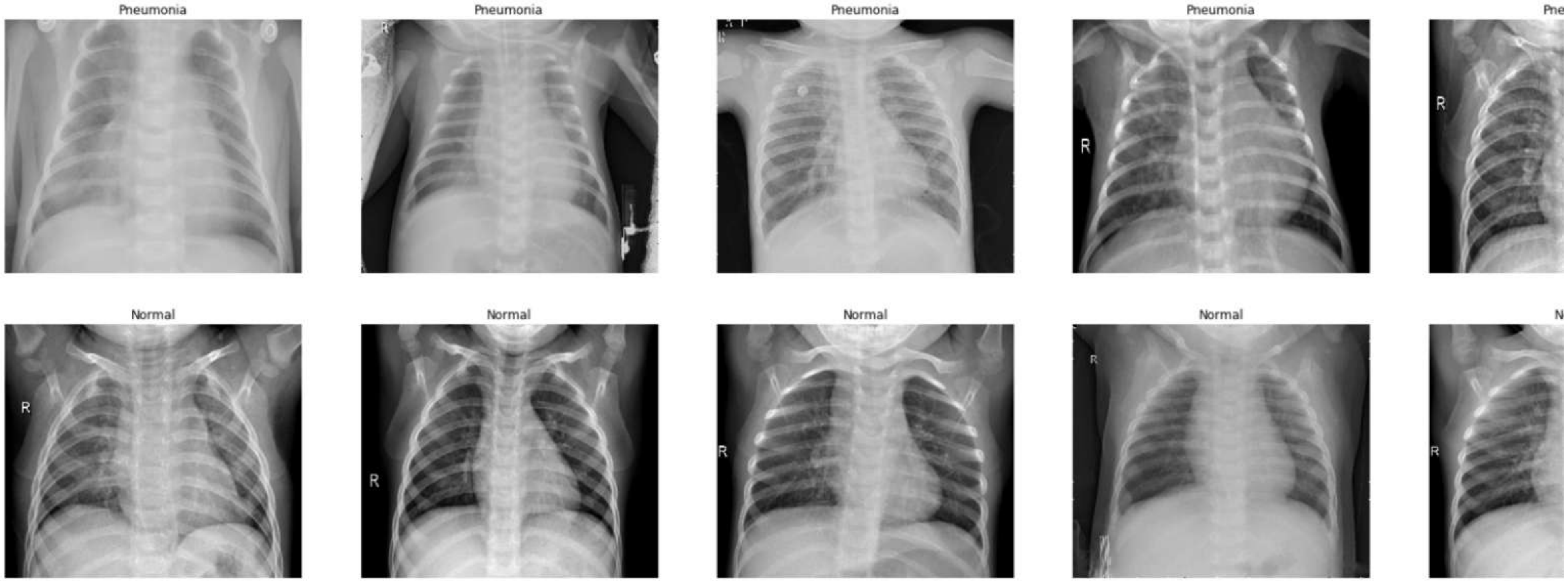
SAINS DATA

Kampus Merdeka INDONESIA JAYA

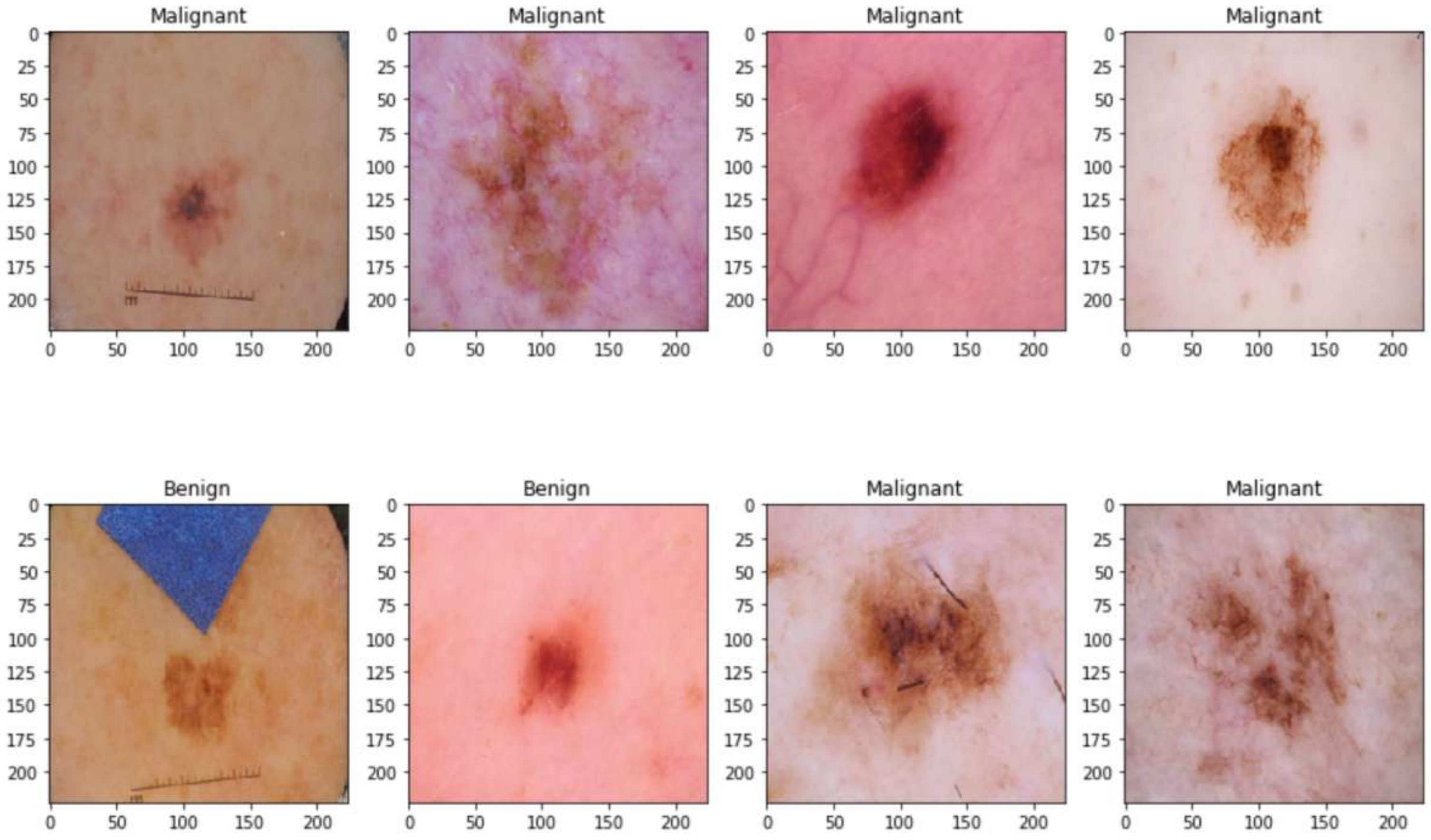
Artificial Intelligence



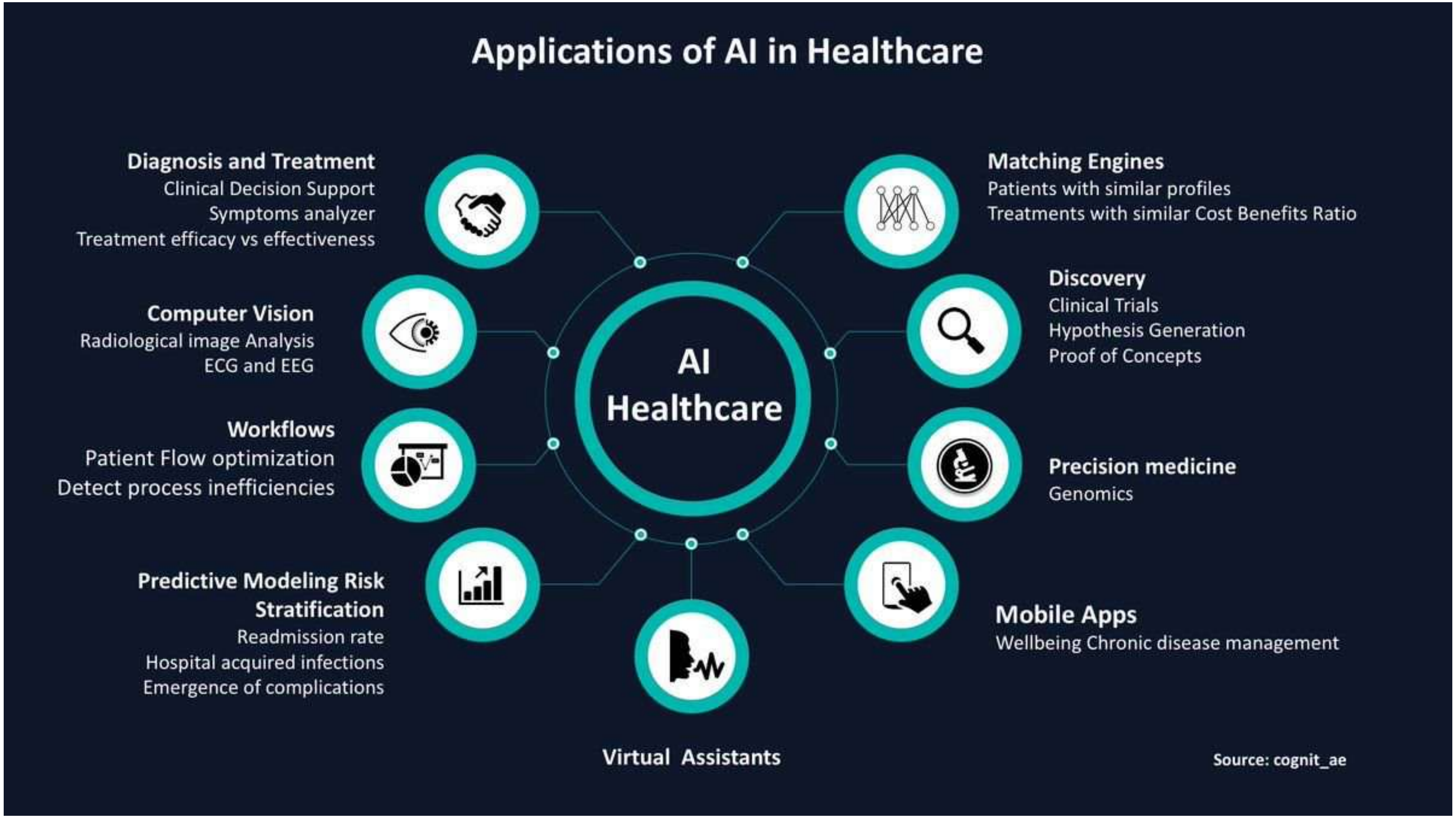
AI in Healthcare: **Detect Pneumonia**



AI in Healthcare: **Skin Cancer Detection**



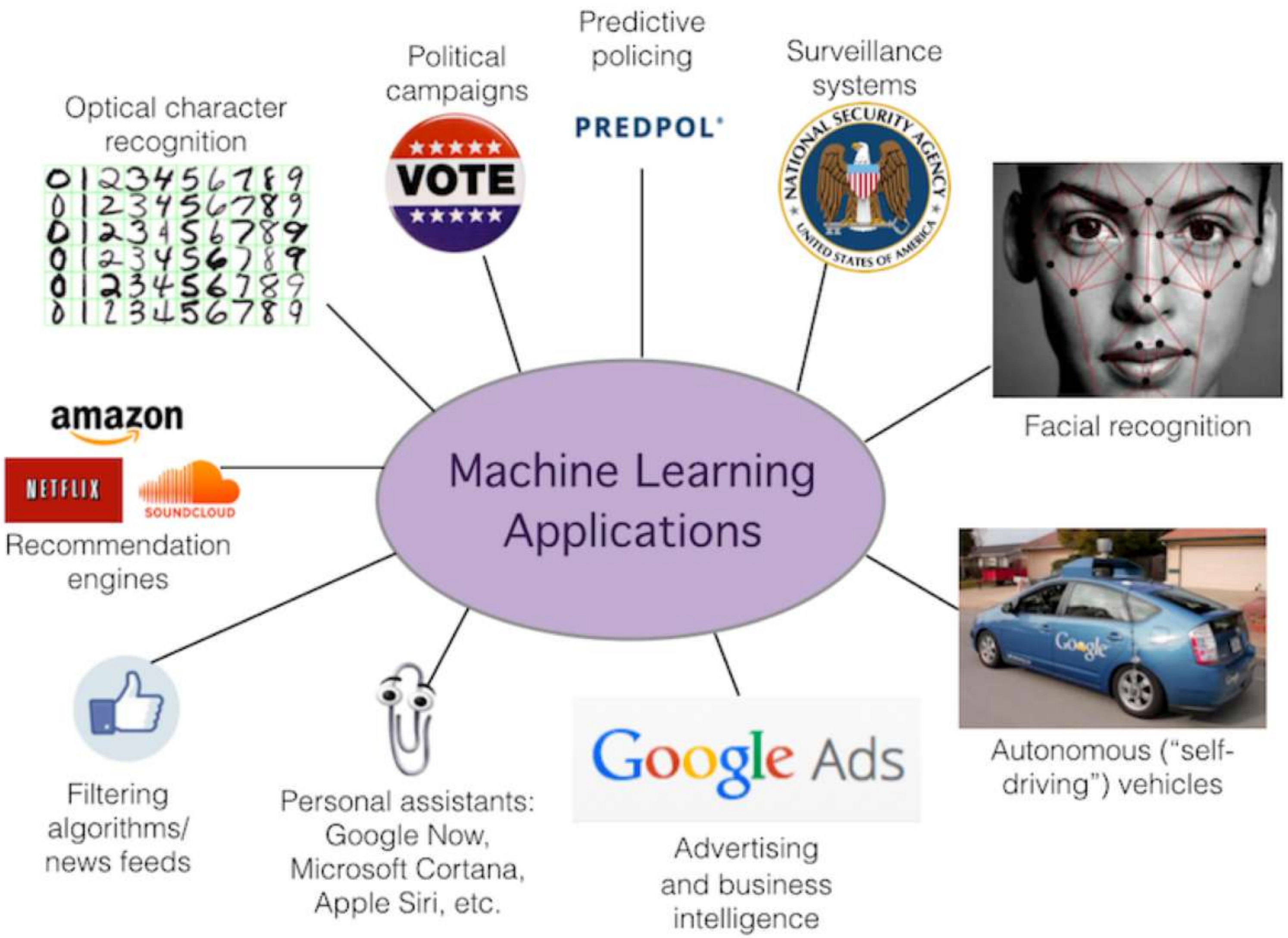
AI in Healthcare



We Focus on **Machine Learning**

- Subset dari Artificial Intelligence
- Konsep Machine Learning: **Performance, Tasks** dan **Experience**

Machine Learning Application

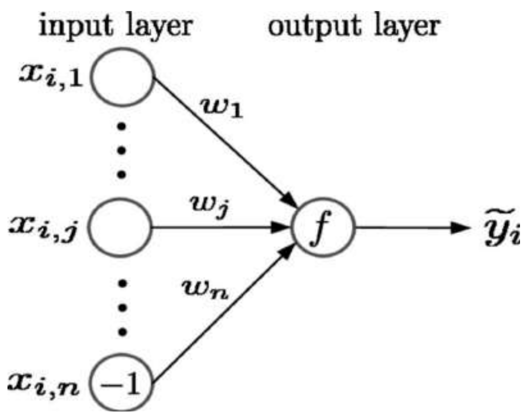


We Need Data!

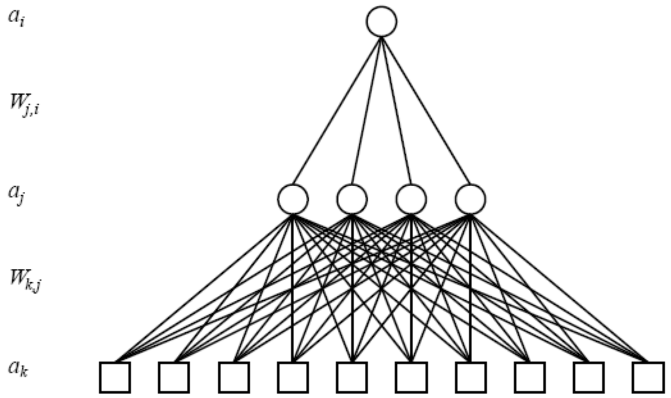
- https://archive.ics.uci.edu/ml/index.php
- https://www.kaggle.com/datasets
- https://data.go.id/

- https://www.kaggle.com/ronitf/heart-disease-uci
- http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

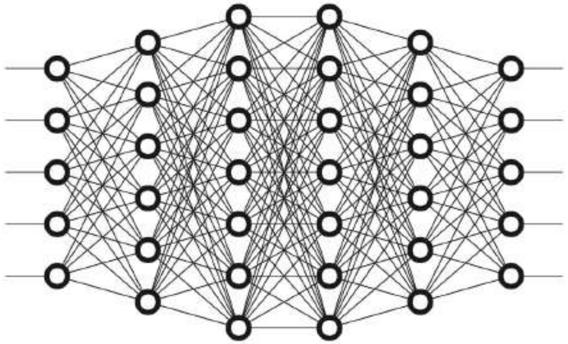
Model ML: Artificial Neural Network (ANN)



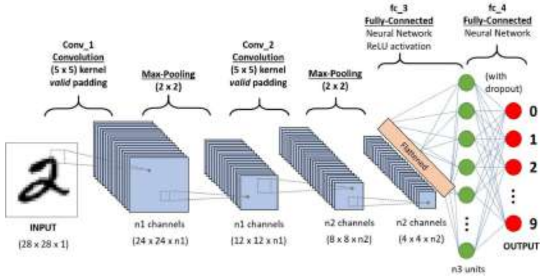
Single Layer Perceptron



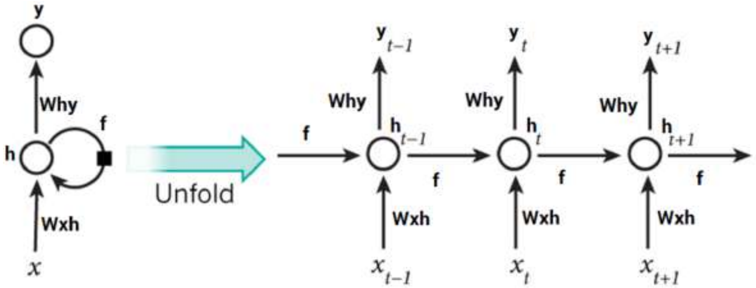
Multi Layer Perceptron



Deep Learning



Convolutional Neural Network
(CNN)



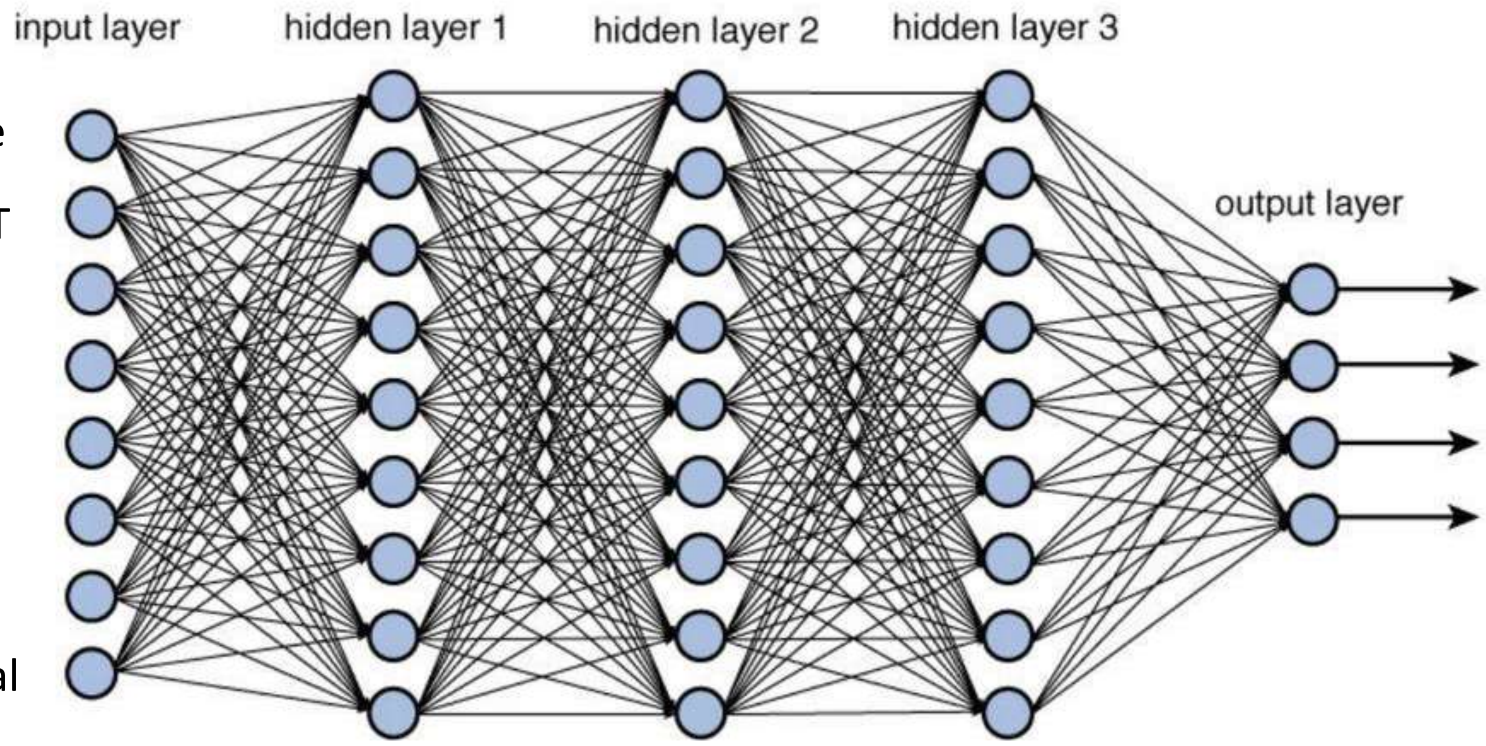
Recurrent Neural Network
(RNN)

Algorithm: Deep Neural Networks

age

CPT

Thal

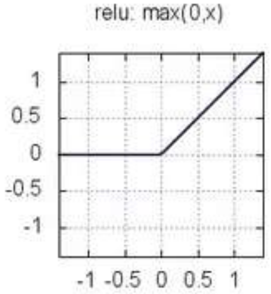


1 = Penyakit Jantung
0 = Tidak

Xi h1i h2i

$$S(x,W) = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} W_{(i-m,j-n)}$$

$$Z(S,U) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij} U_{(i-m,j-n)}$$



$$ReLU = \begin{cases} x, if\ x > 0 \\ 0, otherwise \end{cases}$$

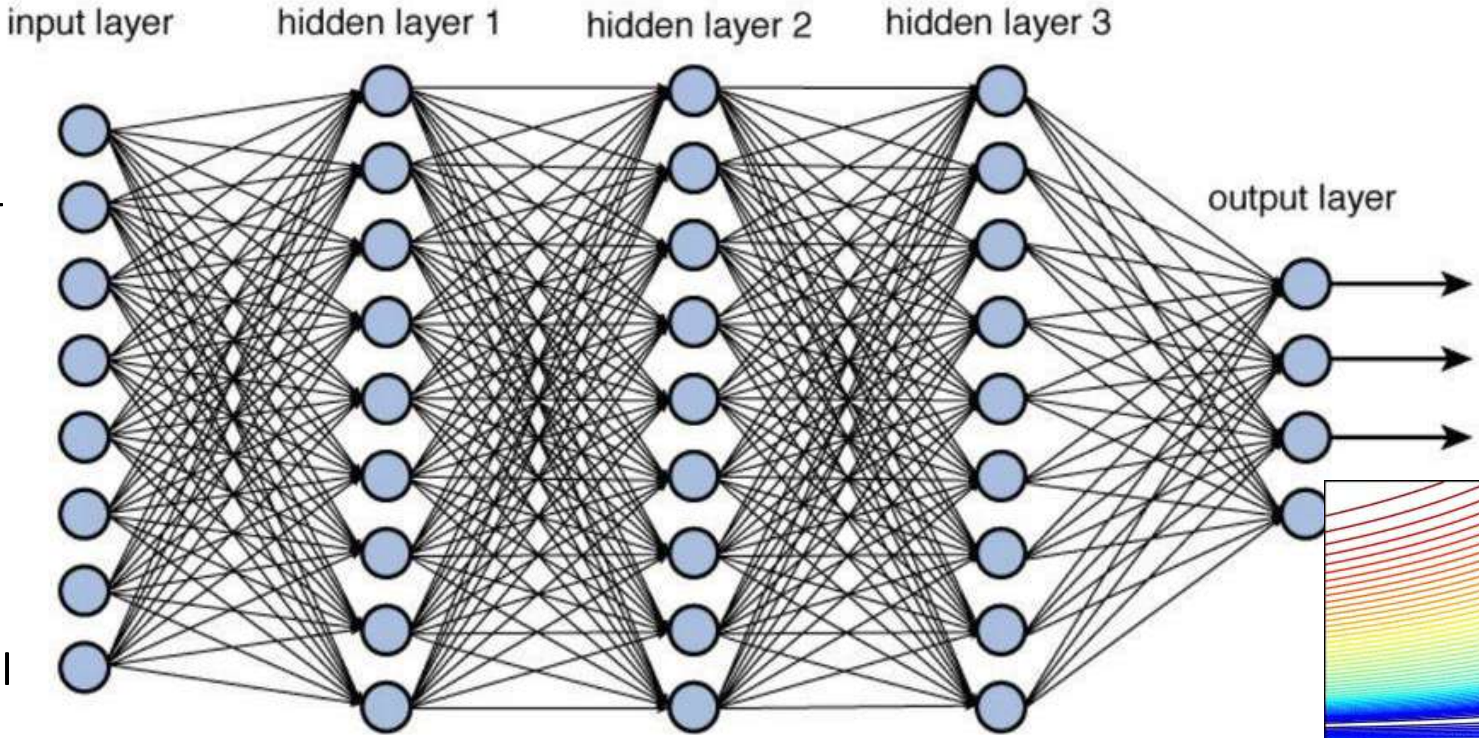
$$Softmax(z_i) = \frac{\exp(Dl(B_{ij} + h_i, W))}{\sum_{i=1}^{n} \exp(Dl(B_{ij} + h_i, W))}$$

Algorithm: Deep Neural Networks

age

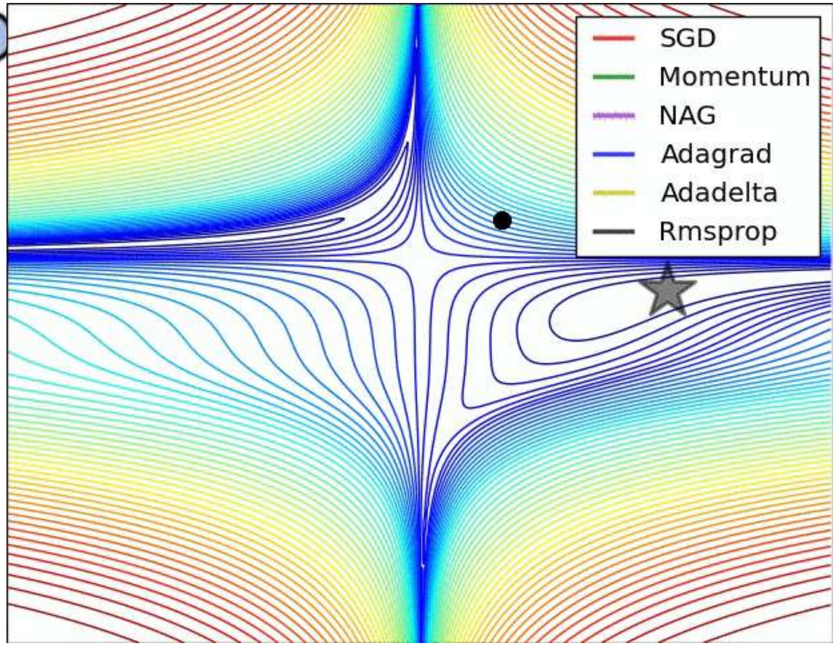
CPT

Thal



Xi h1i h2i

1 = Penyakit Jantung
0 = Tidak

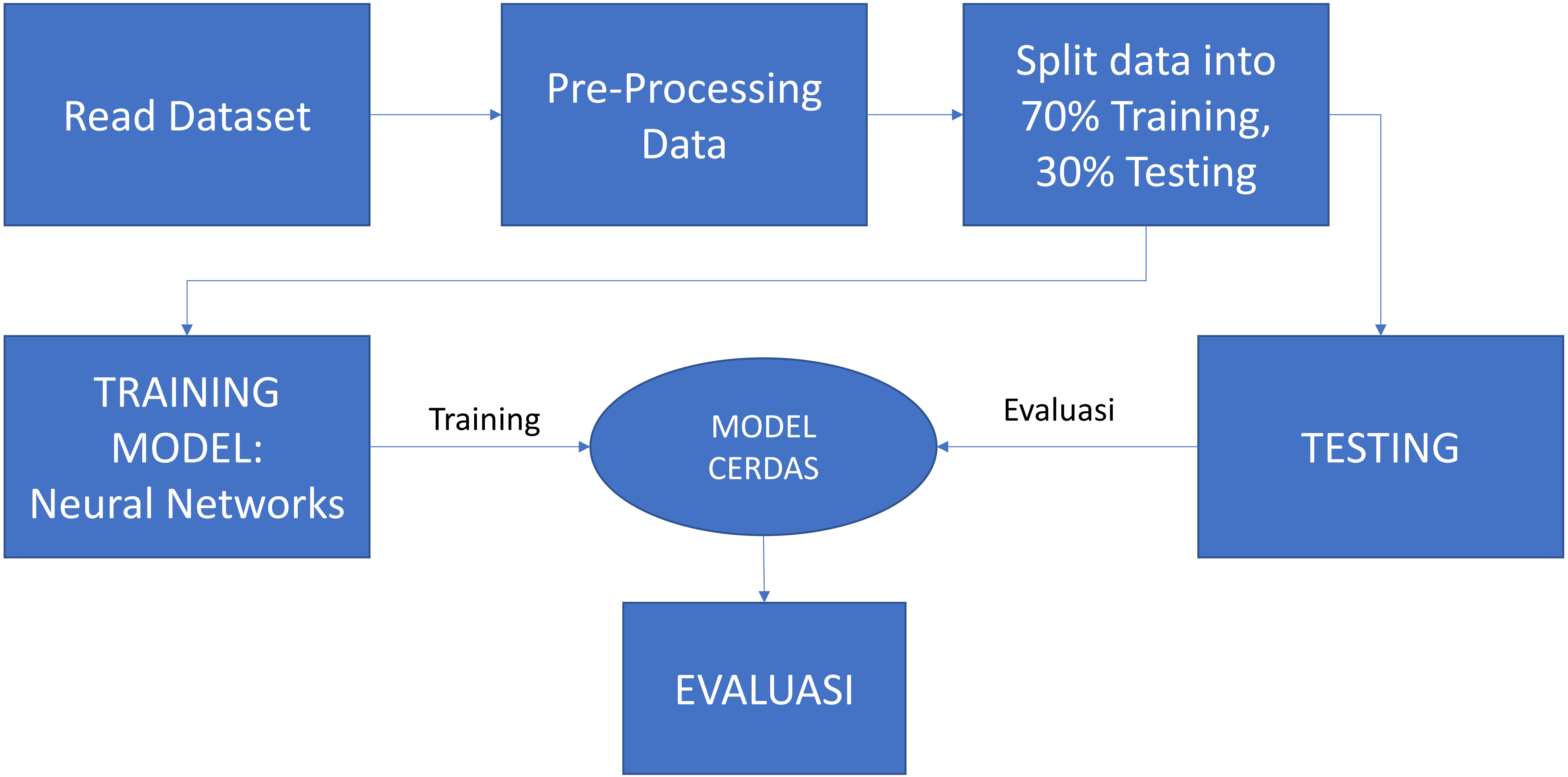


Heart Disease Classification

This database contains 76 attributes, but all published experiments refer **to using a subset of 14 of them**. It is integer valued from 0 (no presence) to 4.

age, sex(1 = male; 0 = female), cpchest pain type, trestbpsresting blood pressure, cholserum cholestoral,
fbs(fasting blood sugar > 120 mg/dl) (1 = true; 0 = false), restecgresting electrocardiographic results
thalachmaximum heart rate achieved, exangexercise induced angina (1 = yes; 0 = no),
oldpeakST depression induced by exercise relative to rest slopethe slope of the peak exercise ST segment
canumber of major vessels (0-3) colored by flourosopy, thal3 = normal; 6 = fixed defect; 7 = reversable defect
target1 or 0

Flow Classification Heart Disease



Create Model

```
model = Sequential()
n_cols = X_train.shape[1]
model.add(Dense(10, activation='relu’,
    input_shape=(n_cols,)))
model.add(Dense(10, activation='relu'))
model.add(Dense(2, activation='softmax’))
model.compile(optimizer='adam’,
    loss='categorical_crossentropy’,
    metrics=['accuracy'])
```

Create Model

```
model.summary()
```

Layer (type)	Output Shape	Param #
dense_16 (Dense)	(None, 10)	140
dense_17 (Dense)	(None, 10)	110
dense_18 (Dense)	(None, 2)	22

Checkpoint Training

- `#Checkpoint in Training`
- `filepath="w{epoch:02d}-{val_acc:.2f}.hdf5"`
- `checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1, save_best_only=True, mode='max')`

- `#Logger`
- `csv_logger = CSVLogger('log.csv', append=True, separator=';')`

Training

- #Training Model
- model.fit(X_train, y_train, validation_split=0.2, epochs=100, callbacks=[csv_logger, checkpoint])

```
Train on 169 samples, validate on 43 samples
Epoch 1/100
169/169 [==============================] - 0s 3ms/step - loss: 0.3526 - acc: 0.8284 - val_loss: 0.4824 - val_acc: 0.7
674

Epoch 00001: val_acc improved from -inf to 0.76744, saving model to w01-0.77.hdf5
Epoch 2/100
169/169 [==============================] - 0s 82us/step - loss: 0.3319 - acc: 0.8698 - val_loss: 0.5189 - val_acc: 0.
7674
```

Model Save / Load

#Model Save

- `model.save("model.h5")`

#Model Load

- `model_baru = load_model('model.h5')`
- `model_baru.summary()`

Testing

```
y_pred = model.predict_classes(X_test)
y_test_ = np.argmax(y_test, axis=1)
print(classification_report(y_test_,
y_pred, target_names=["Sakit","Tidak"]))
```

```
             precision    recall  f1-score   support

      Sakit       0.77      0.80      0.79        41
      Tidak       0.83      0.80      0.82        50

avg / total       0.80      0.80      0.80        91
```

Model Evaluation

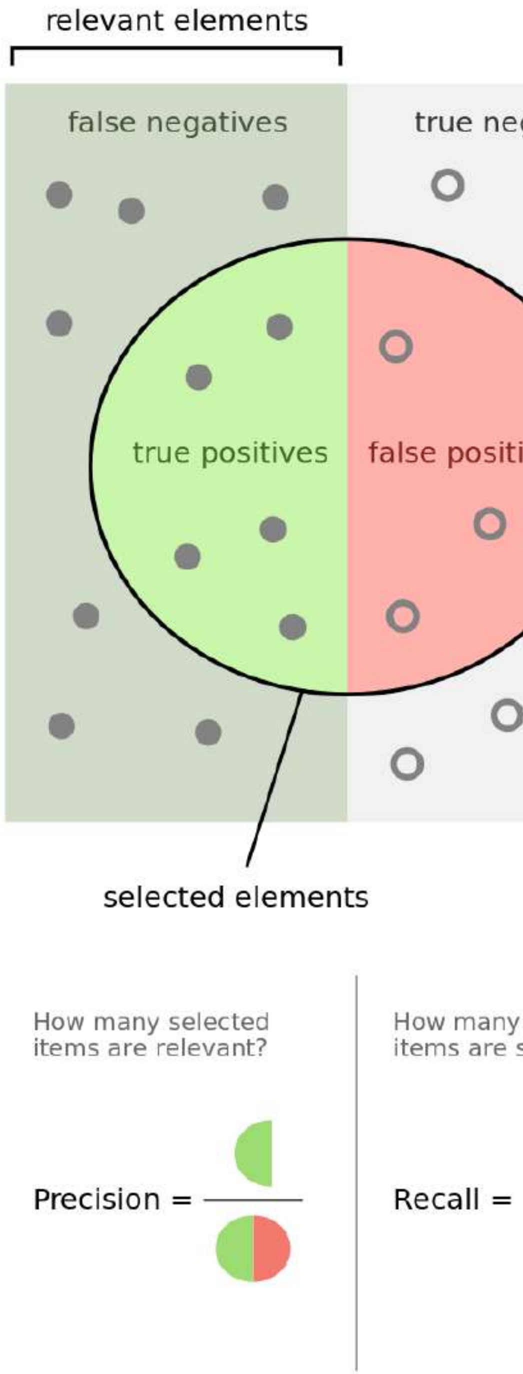
- Precision $(P) = \frac{|Relevan \cap Retrieved|}{|Relevant|}$
- Recall $(R) = \frac{|Relevan \cap Retrieved|}{|Retreived|}$
- F-Score $(F) = 2\frac{P \,.\, R}{P+R}$
- $Kappa = \frac{P(A)-P(E)}{P(E)-1}$

Precision Recall + Confusion Matrix

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Recall} = \frac{tp}{tp + fn} \qquad F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

```
[[249,   0,   8,   0,  10,   0,   7,   4,   0],
 [  0, 261,   4,   0,   0,   0,   0,   1,   4],
 [ 15,   3, 232,   0,   1,   0,   0,   2,   0],
 [  0,   0,   0, 363,   0,   7,   1,   0,   0],
 [ 63,   1,   7,  16,  14,   5,  13,  12,   0],
 [  1,   0,   0,  35,   1,  15,  11,   0,   0],
 [  0,   0,   0,   0,   0,   0, 393,   1,   0],
 [  2,   0,   0,   0,   0,   0,   2, 514,   0],
 [  0,  55,   2,   0,   0,   0,   0,   0,  50]])
```

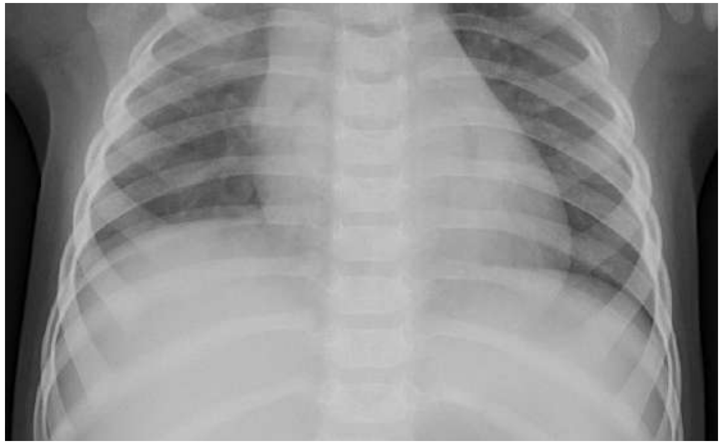


Medical Image Classification

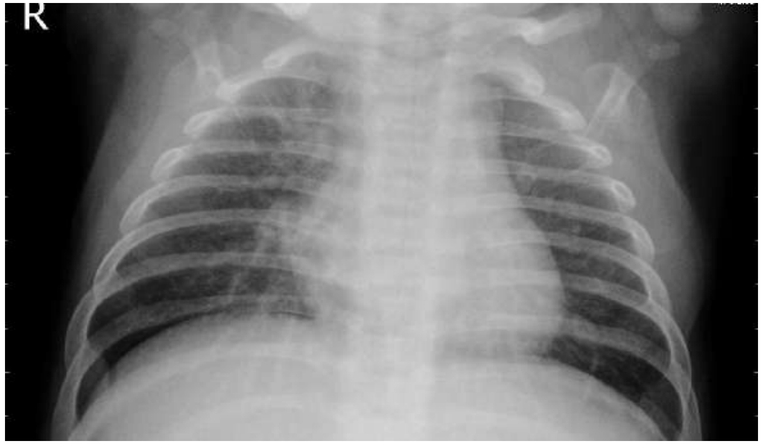
Pneumonia Image Classification

This database contains of X-Ray images normal Chest, Pneumonia caused by virus and Pneumonia caused by bacteria.

Normal

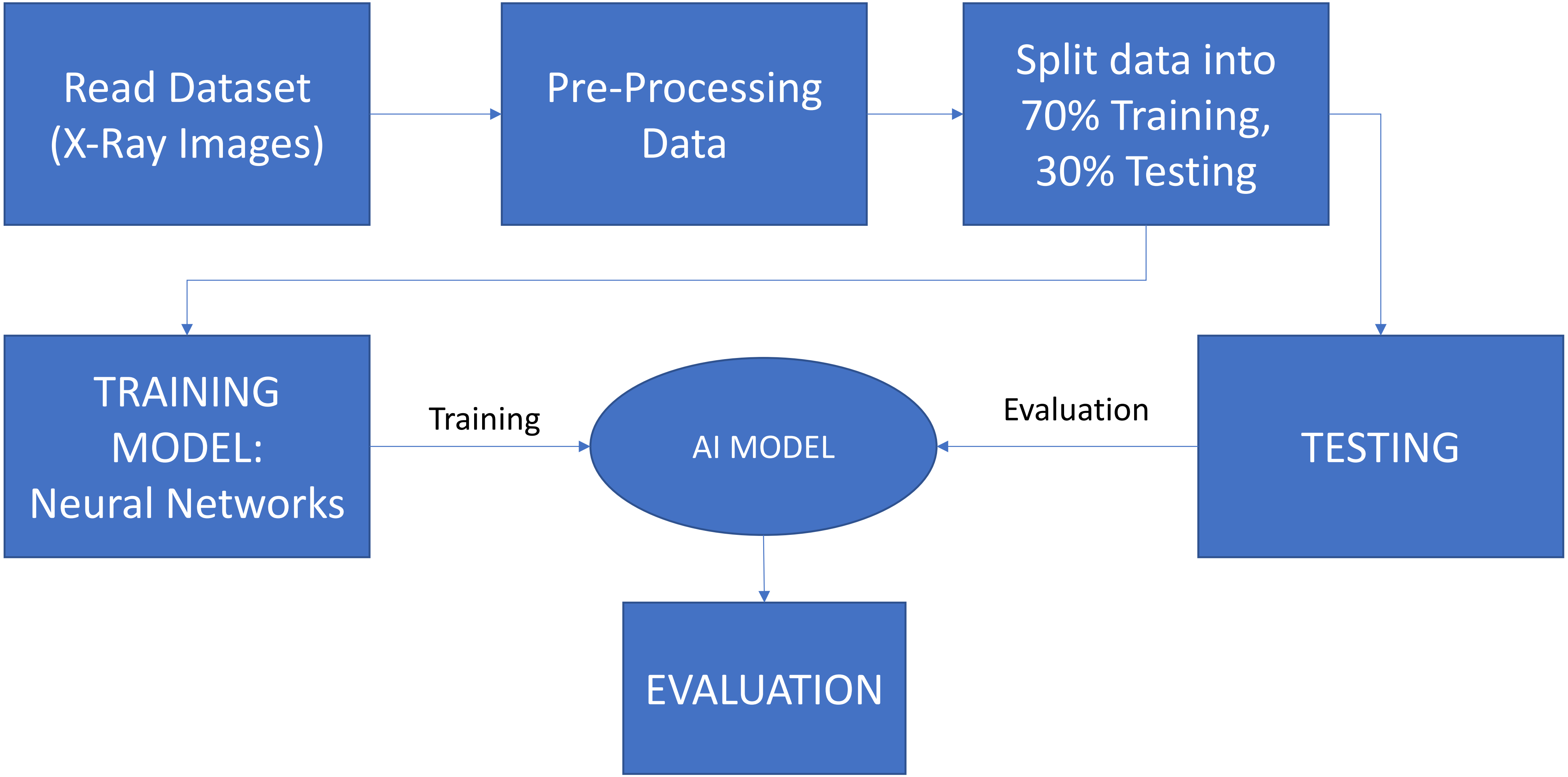


Bacteria



Virus

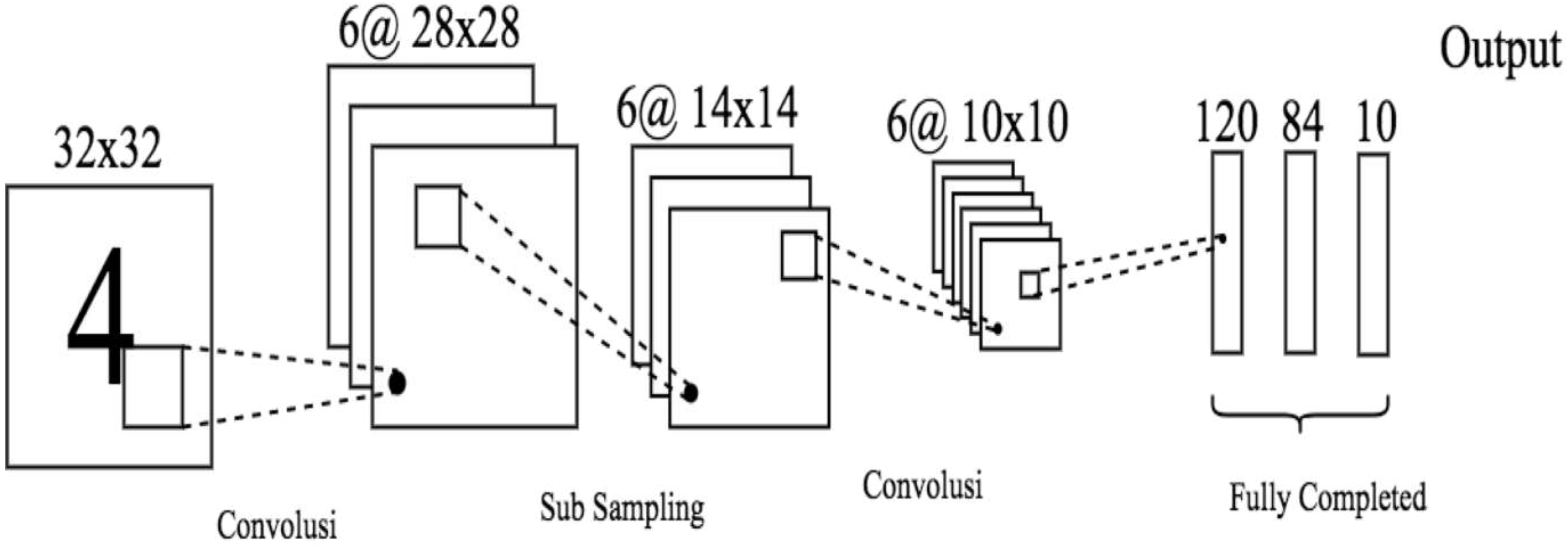
Flow Classification Pneumonia Disease



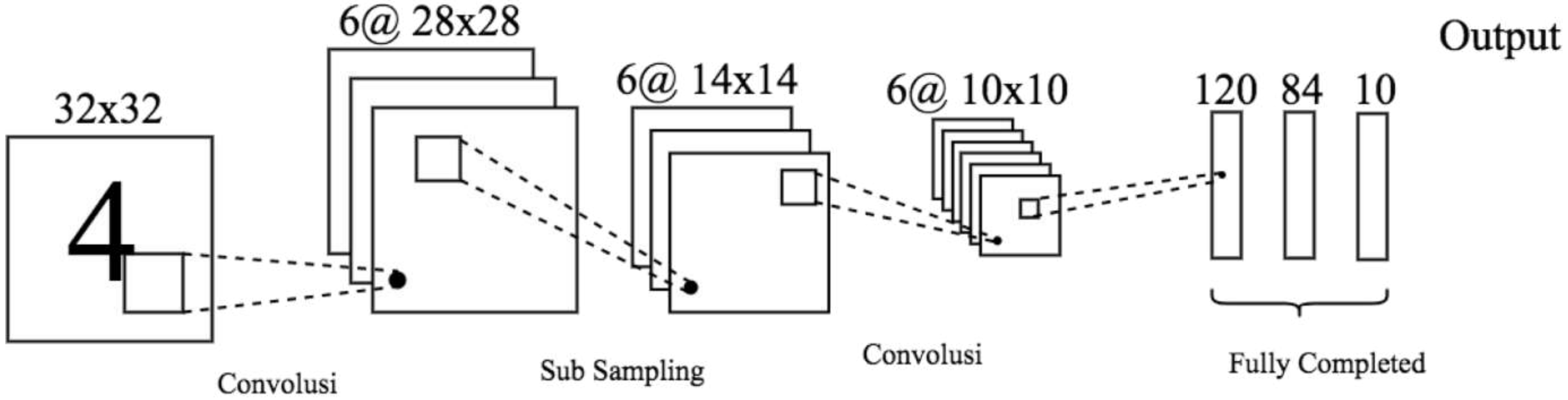
In order to classify images, we Need variant of Neural Networks for Image Classification!

CONVOLUTIONAL NEURAL NETWORKS

Deep Convolutional Neural Networks



Convolutional Neural Networks



Rs, Pa, PS, CR, PS …

$$S(x,W) = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}W_{(i-m,j-n)}\ ;\ Z(S,U) = \sum_{i=1}^{n}\sum_{j=1}^{m} S_{ij}U_{(i-m,j-n)}\ ;\ Softmax(z_i) = \frac{\exp(Z_i)}{\sum_{i=1}^{n}\exp(Z_j)}$$

Convolution Example

$$I = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{bmatrix} \star K = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} (I \star K)(0,0) = \;& I(0,0)K(0,0) + I(0,1)K(0,1) + I(0,2)K(0,2) + \\ & I(1,0)K(1,0) + I(1,1)K(1,1) + I(1,2)K(1,2) + \\ & I(2,0)K(2,0) + I(2,1)K(2,1) + I(2,2)K(2,2) \\ = \;& 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + \\ & 0 \cdot 0 + 0 \cdot 2 + 0 \cdot 0 + \\ & 0 \cdot 1 + 1 \cdot 1 + 2 \cdot 0 \\ = \;& 1 \end{aligned}$$

$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m, j+n)$$

Convolution Example

Can be viewed as a "sliding window" operation:

K

I

1_0	0_0	0_1	1	2
0_0	0_2	0_0	3	0
0_1	1_1	2_0	1	1
1	1	3	0	0
3	0	0	0	1

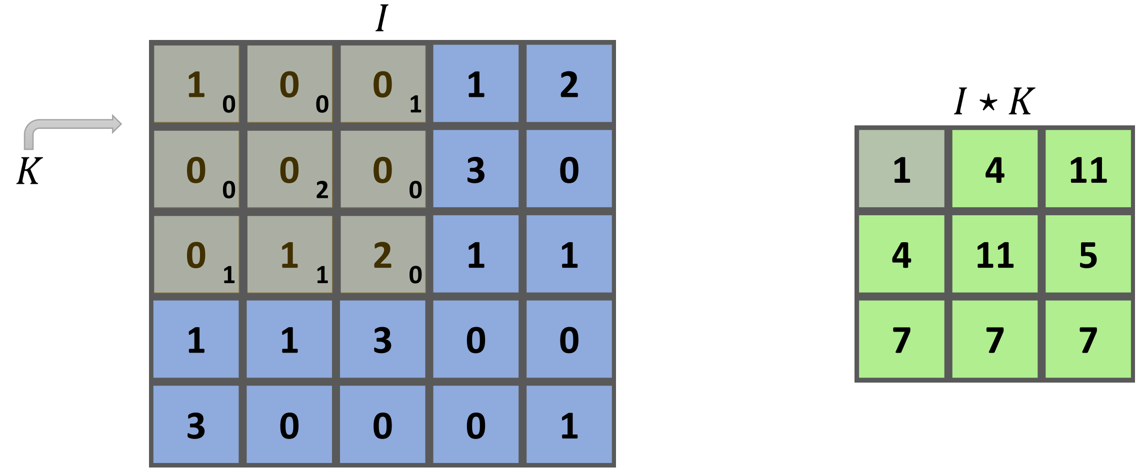
$I \star K$

1	4	11
4	11	5
7	7	7

$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

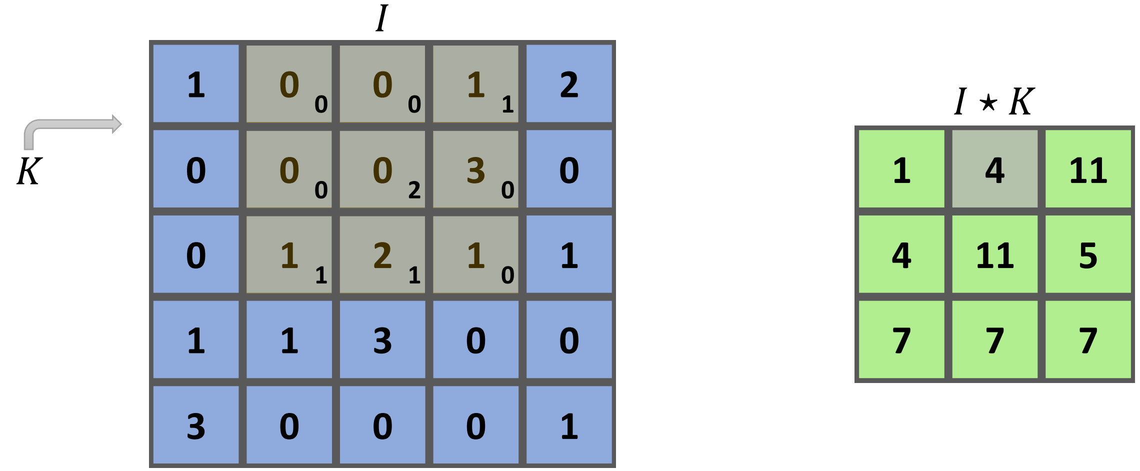
Can be viewed as a “sliding window” operation:



$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a “sliding window” operation:



$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a "sliding window" operation:

K

I

1	0	0_0	1_0	2_1
0	0	0_0	3_2	0_0
0	1	2_1	1_1	1_0
1	1	3	0	0
3	0	0	0	1

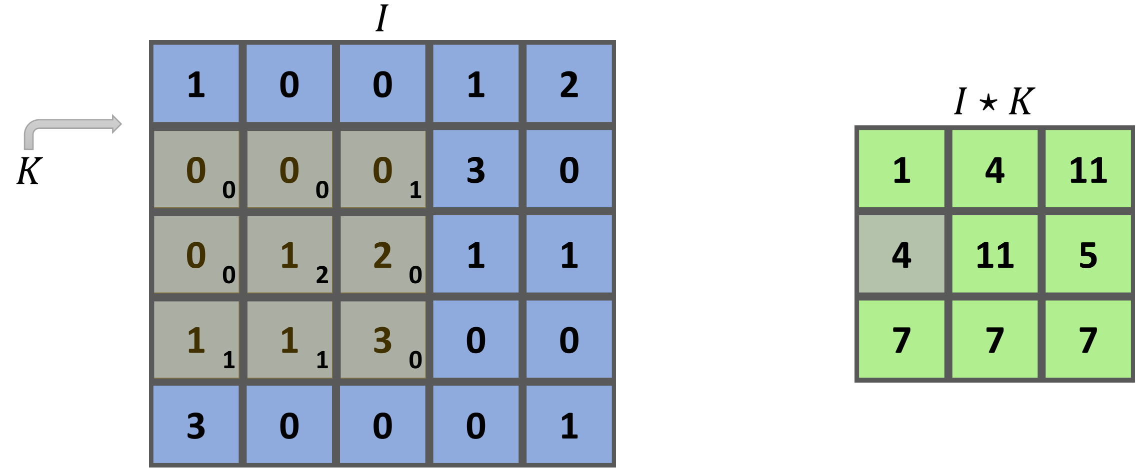
$I \star K$

1	4	11
4	11	5
7	7	7

$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a "sliding window" operation:



$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a “sliding window” operation:

I

K

1	0	0	1	2
0	0_{0}	0_{0}	3_{1}	0
0	1_{0}	2_{2}	1_{0}	1
1	1_{1}	3_{1}	0_{0}	0
3	0	0	0	1

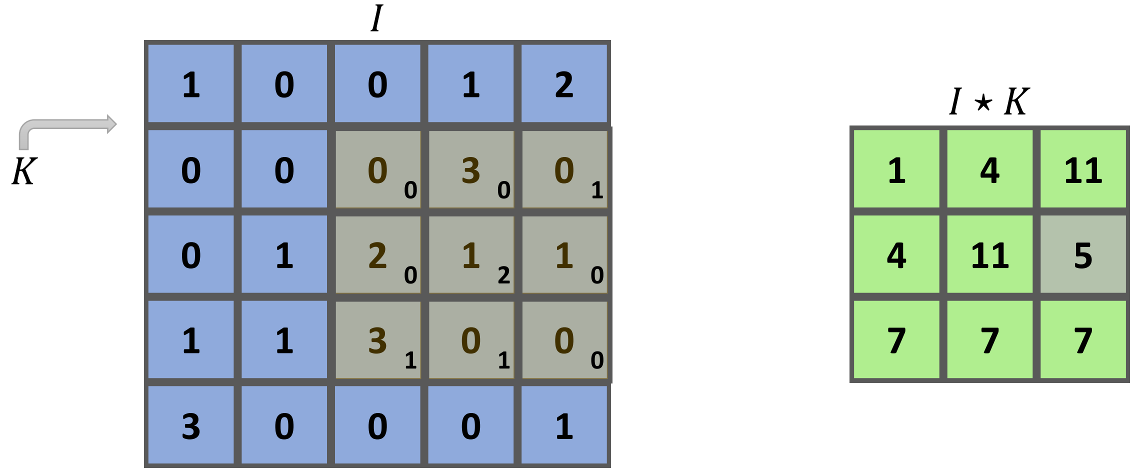
$I \star K$

1	4	11
4	11	5
7	7	7

$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a "sliding window" operation:



$$(I \star K)(i,j) == \sum_{m} \sum_{n} I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a "sliding window" operation:

K

I

1	0	0	1	2
0	0	0	3	0
0_{0}	1_{0}	2_{1}	1	1
1_{0}	1_{2}	3_{0}	0	0
3_{1}	0_{1}	0_{0}	0	1

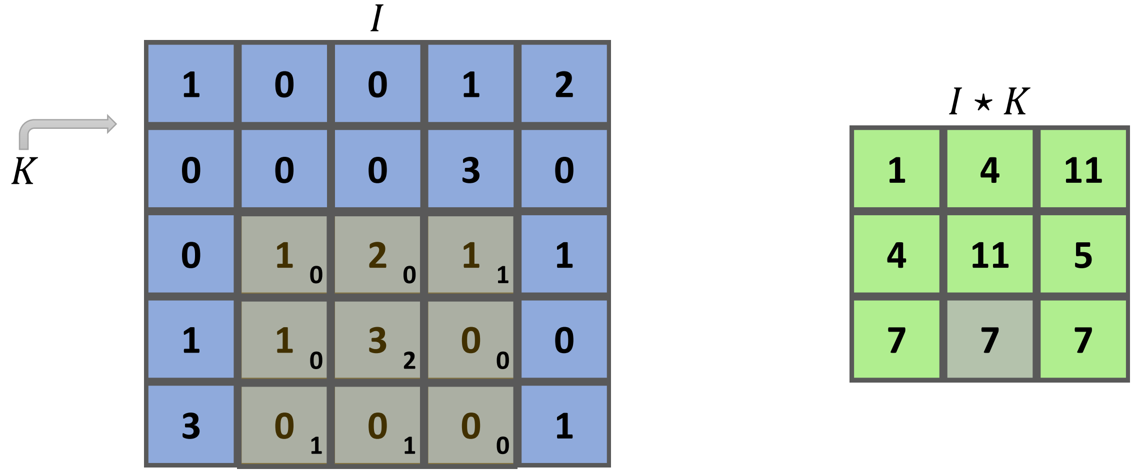
$I \star K$

1	4	11
4	11	5
7	7	7

$$(I \star K)(i,j) == \sum_{m} \sum_{n} I(m,n)K(i+m, j+n)$$

Convolution Example

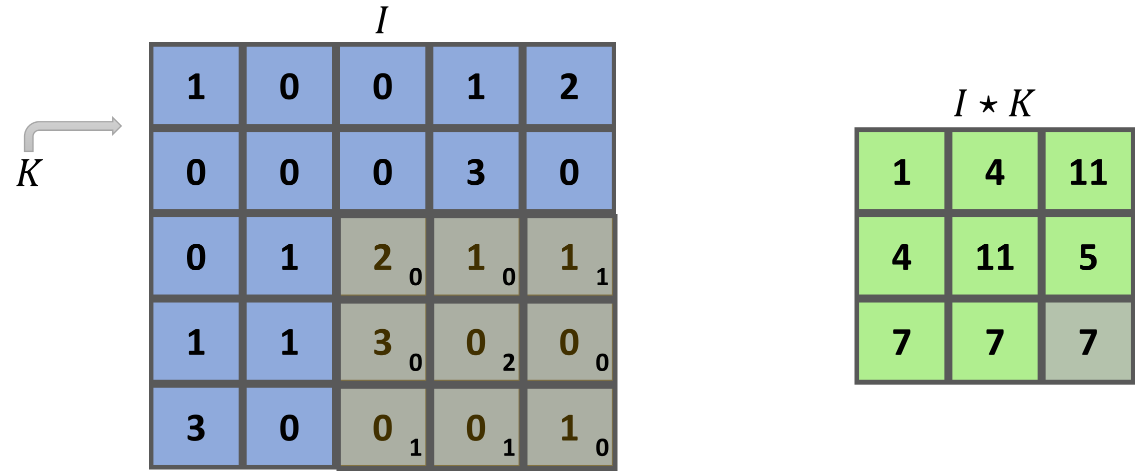
Can be viewed as a “sliding window” operation:



$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

Convolution Example

Can be viewed as a “sliding window” operation:



$$(I \star K)(i,j) == \sum_m \sum_n I(m,n)K(i+m,j+n)$$

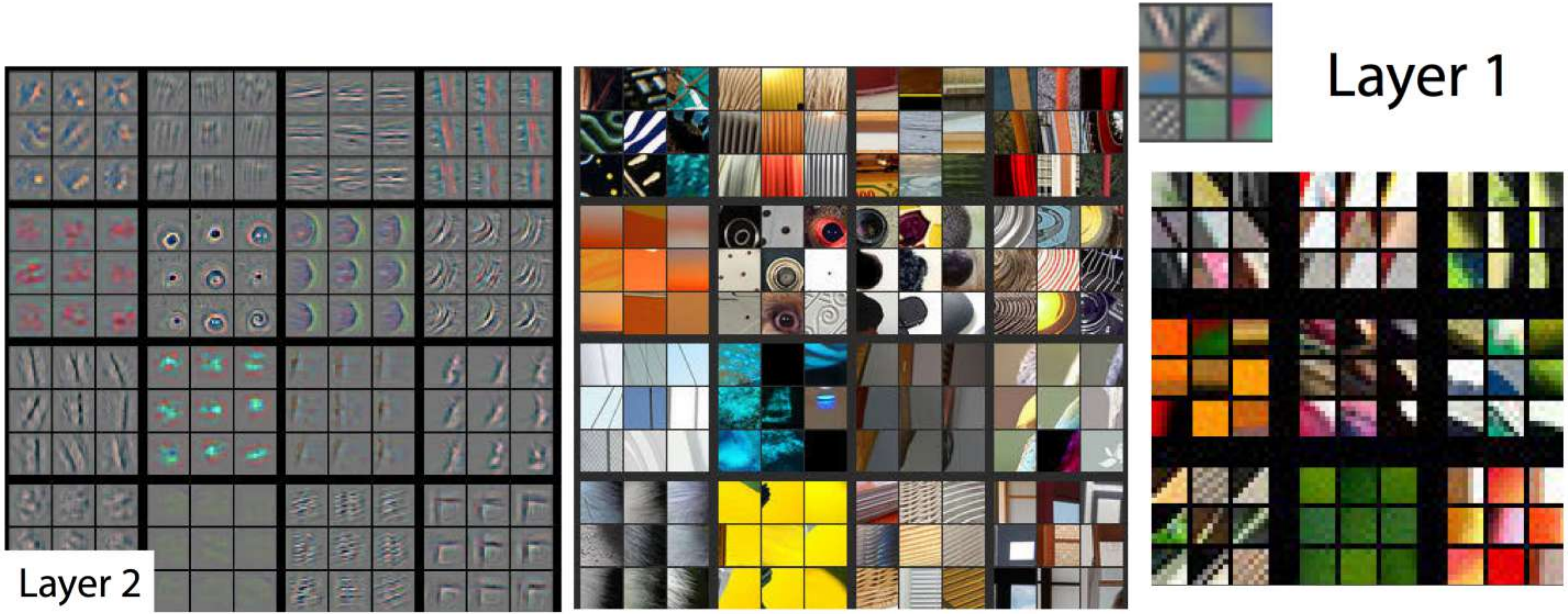
Convolution Example

$$
\begin{bmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 & 0 \\ 3 & 0 & 0 & 0 & 1 \end{bmatrix} \star \begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 11 \\ 4 & 11 & 5 \\ 7 & 7 & 7 \end{bmatrix}
$$

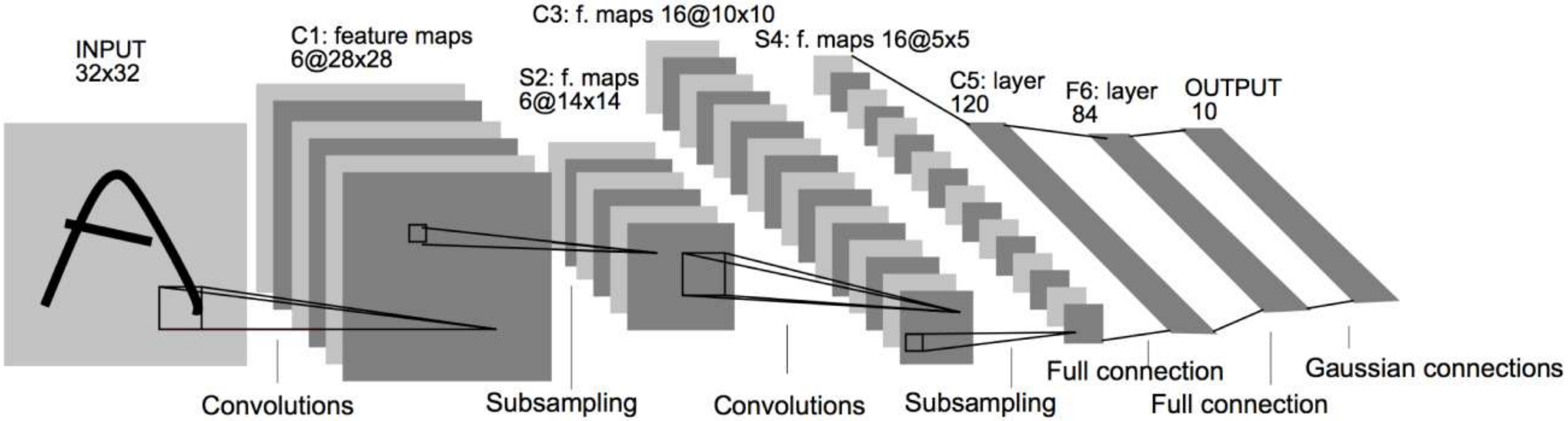
Often called the feature map

$$
(I \star K)(i,j) == \sum_{m} \sum_{n} I(m,n) K(i+m, j+n)
$$

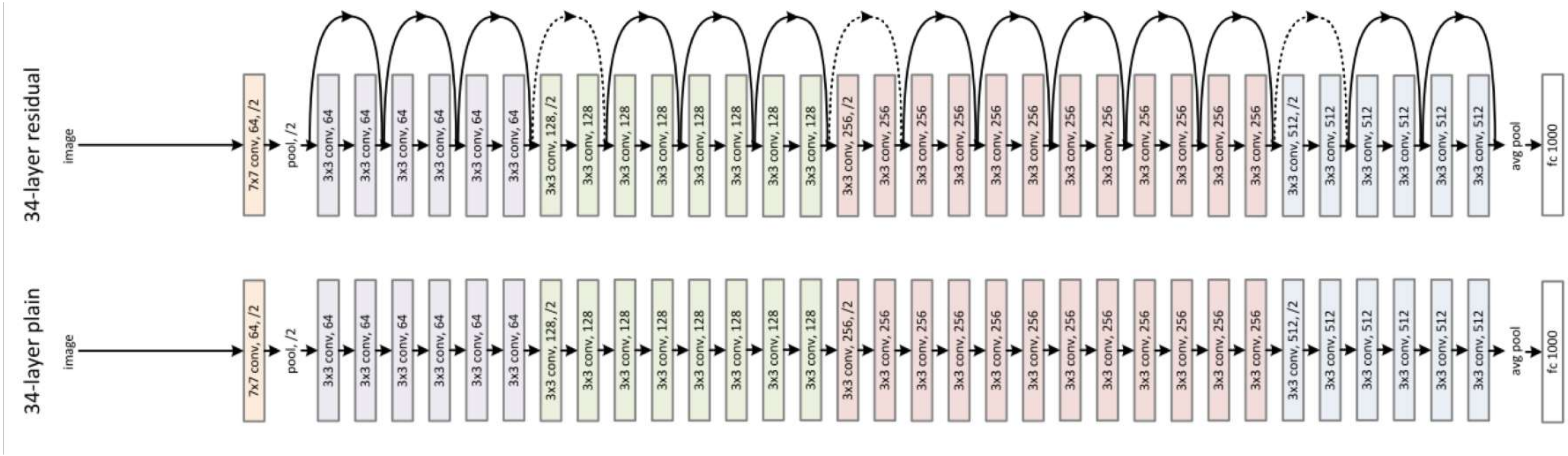
Feature Visualization



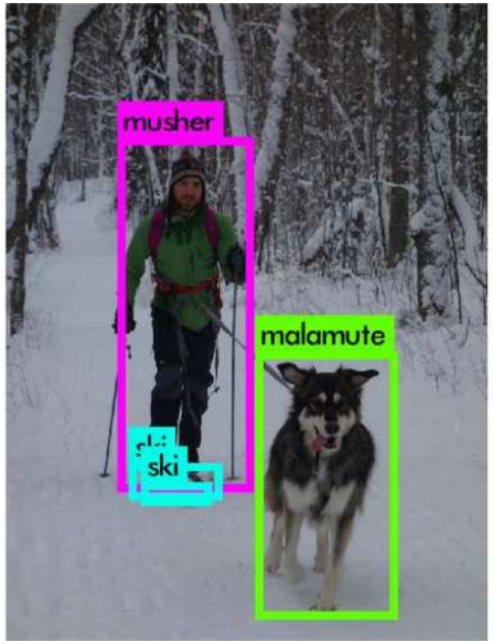
Variant CNN: AlexNet

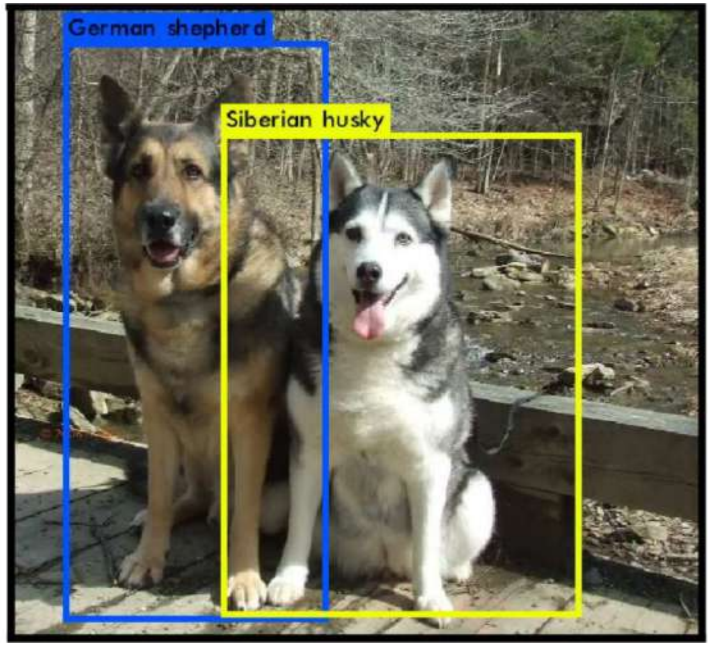


Variant CNN: ResNet

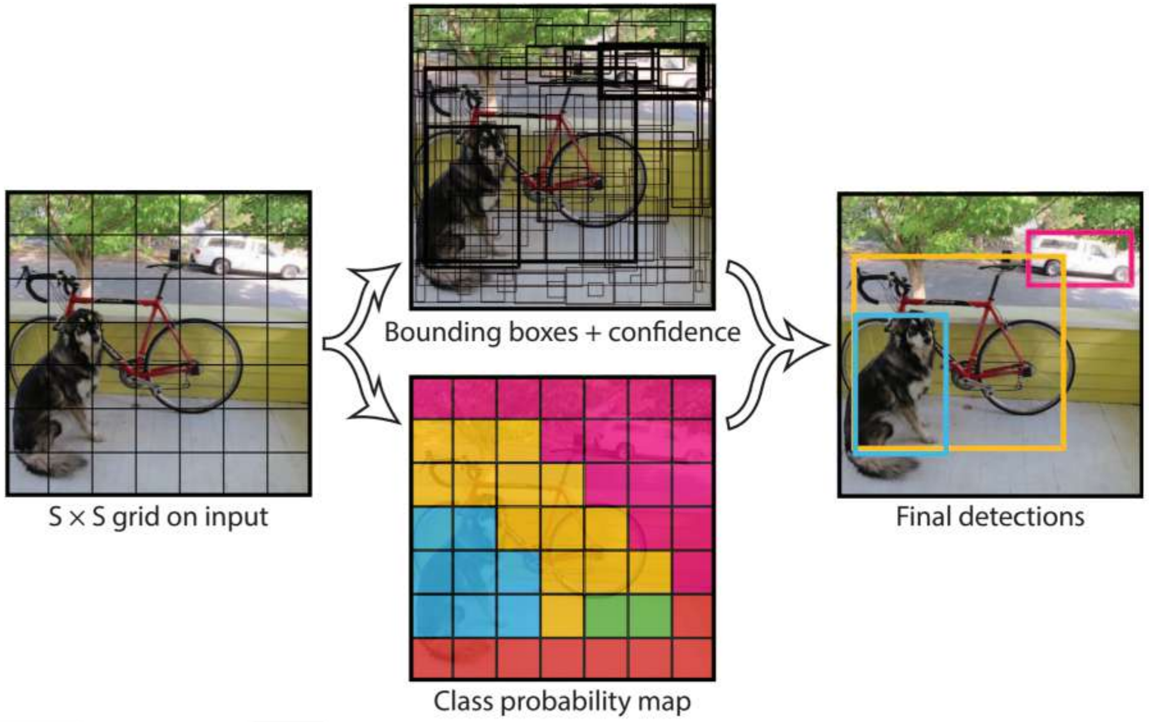


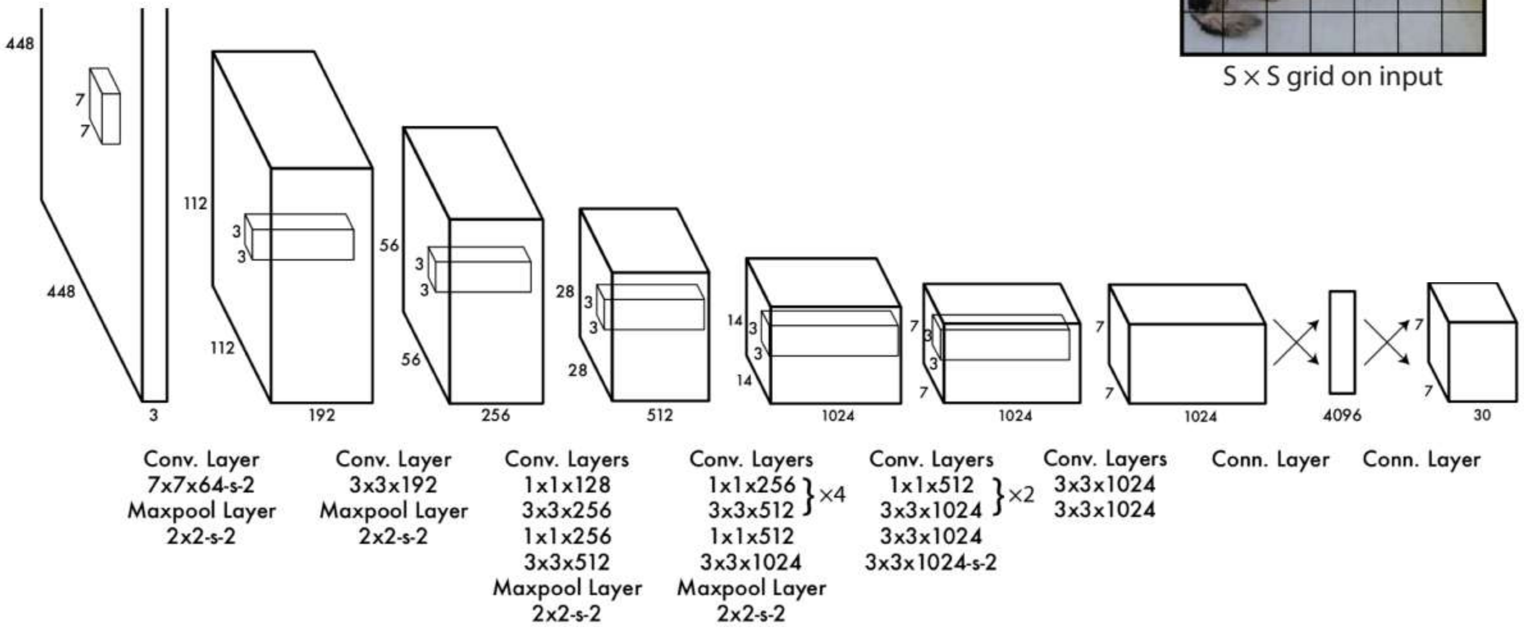
Variant CNN: Object Detection





Variant CNN: YOLO





Read Dataset

```
path = os.path.abspath('chestxray.ipynb')
path = re.sub('[a-zA-Z\s._]+$', '', path)
dirs = os.listdir(path+'../dataset/xray/')
label = 0
X = []
y = []
example_data = []
example_label = []
```

Read Dataset

```
for i in dirs: #loop all directory
    count = 0 #Temporary variable count
    for pic in glob.glob(path+'/*.jpeg’):
         im = cv2.imread(pic)
        im = cv2.resize(im,(100,100))
        im = np.array(im)
        count = count + 1
        X.append(im)
        y.append(label)
```

Read Dataset

```
          if(count==3):
              example_data.append({str(i):im})
              example_label.append(i)
               label = label + 1
               X = np.array(X)
               y = np.array(y)
```

Data Split

```
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.33,
random_state=42)


X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
y_train = to_categorical(y_train, 3)
y_test = to_categorical(y_test, 3)
```

Model Creation

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=(100,100,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(9, activation='softmax'))
```

Model Compile

```
epochs = 25
lrate = 0.01
decay = lrate/epochs
sgd = SGD(lr=lrate, momentum=0.9, decay=decay, nesterov=False)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
print(model.summary())
```

Model Training

```
model.fit(X_train, y_train,
validation_data=(X_test, y_test), epochs=epochs,
batch_size=32)

scores = model.evaluate(X_test, y_test, verbose=0)

print("Accuracy: %.2f%%" % (scores[1]*100))
```

Model Evaluation

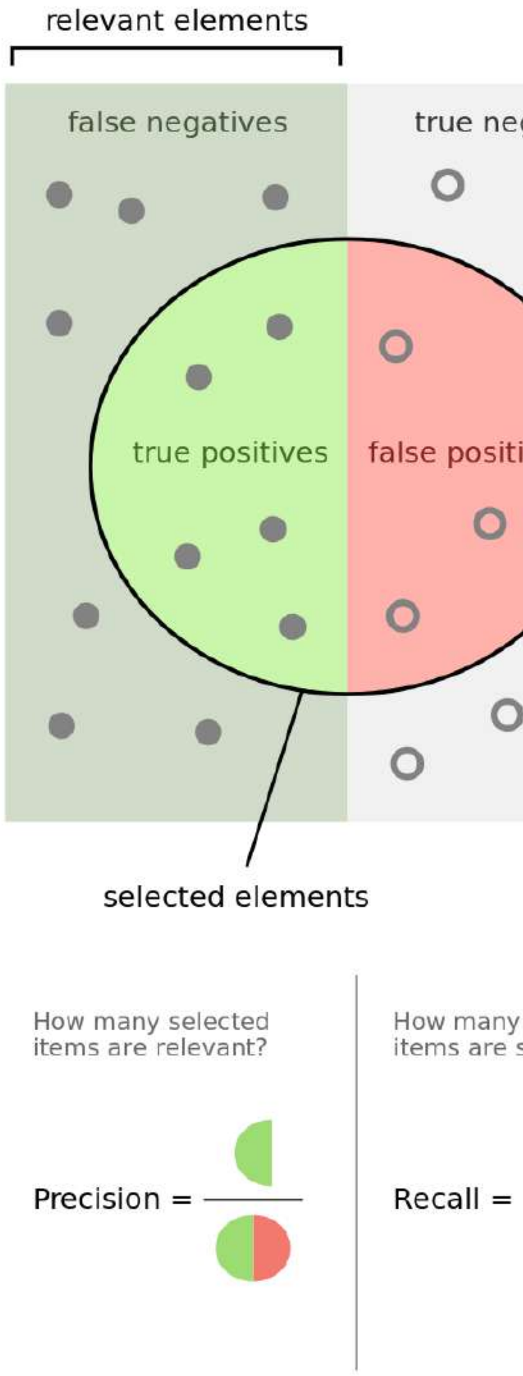
```
model = model.save('my_model.h5')
model_baru = load_model('my_model.h5')
y_pred = model_baru.predict_classes(X_test)
y_test_ = np.argmax(y_test, axis=1)
print(classification_report(y_test_, y_pred,
target_names=mapping.values()))
```

Precision Recall + Confusion Matrix

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

$$\text{Recall} = \frac{tp}{tp + fn} \qquad F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

```
[[249,   0,   8,   0,  10,   0,   7,   4,   0],
 [  0, 261,   4,   0,   0,   0,   0,   1,   4],
 [ 15,   3, 232,   0,   1,   0,   0,   2,   0],
 [  0,   0,   0, 363,   0,   7,   1,   0,   0],
 [ 63,   1,   7,  16,  14,   5,  13,  12,   0],
 [  1,   0,   0,  35,   1,  15,  11,   0,   0],
 [  0,   0,   0,   0,   0,   0, 393,   1,   0],
 [  2,   0,   0,   0,   0,   0,   2, 514,   0],
 [  0,  55,   2,   0,   0,   0,   0,   0,  50]])
```



Result

	precision	recall	f1-score	support
NORMAL	0.86	0.87	0.86	83
PNEUMONIA	0.91	0.90	0.91	123
accuracy			0.89	206
macro avg	0.88	0.88	0.88	206
weighted avg	0.89	0.89	0.89	206

Exercise: Skin Cancer Classification

Read skin cancer dataset using the similar techniques as chest x-ray. Classify the image whether the sample images benign or malignant.

Use appropriate techniques like image preprocessing or feature selection to improve your model

The final result is **we can create the model that can classify the image whether malignant or benign**