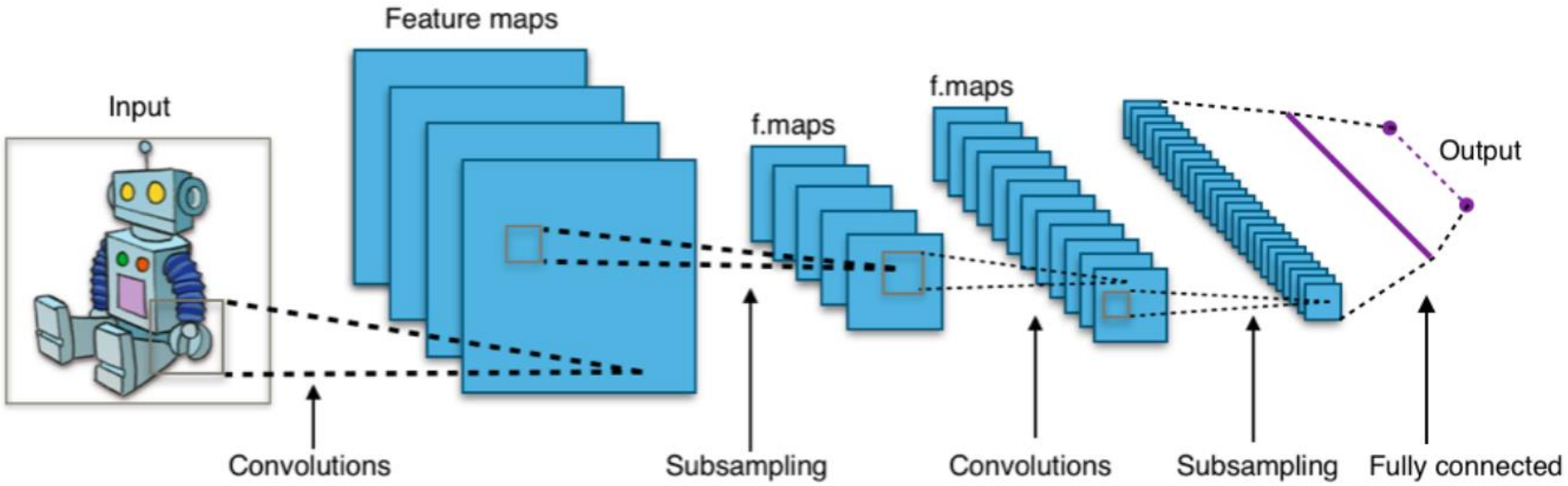
Introduction to CNN

Abdul Haseeb

BS(AI)-III

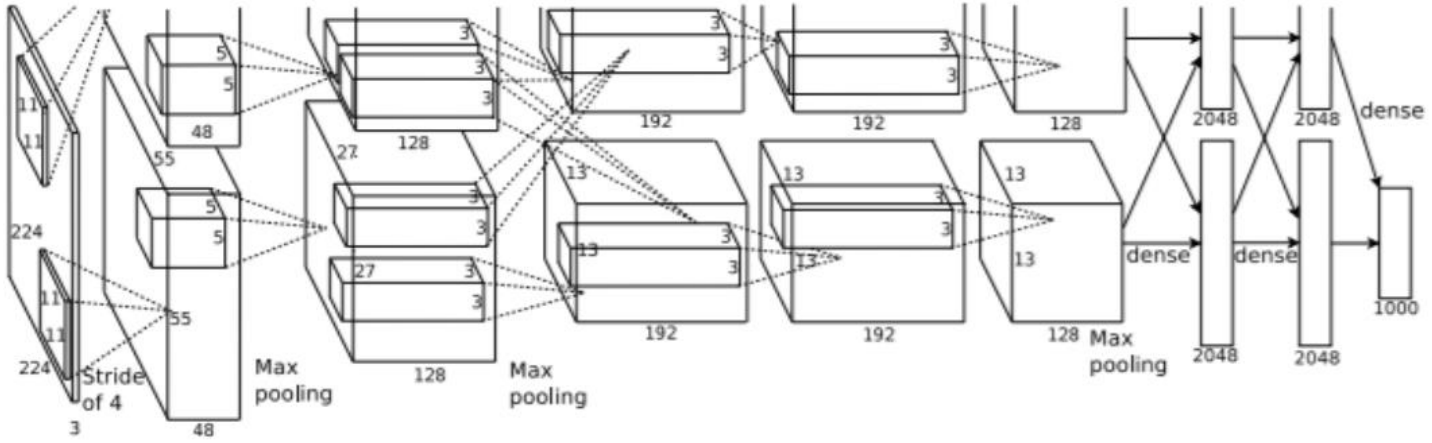
Convolutional Neural Networks

- A class of Neural Networks
  - Takes image as input (mostly)
  - Make predictions about the input image

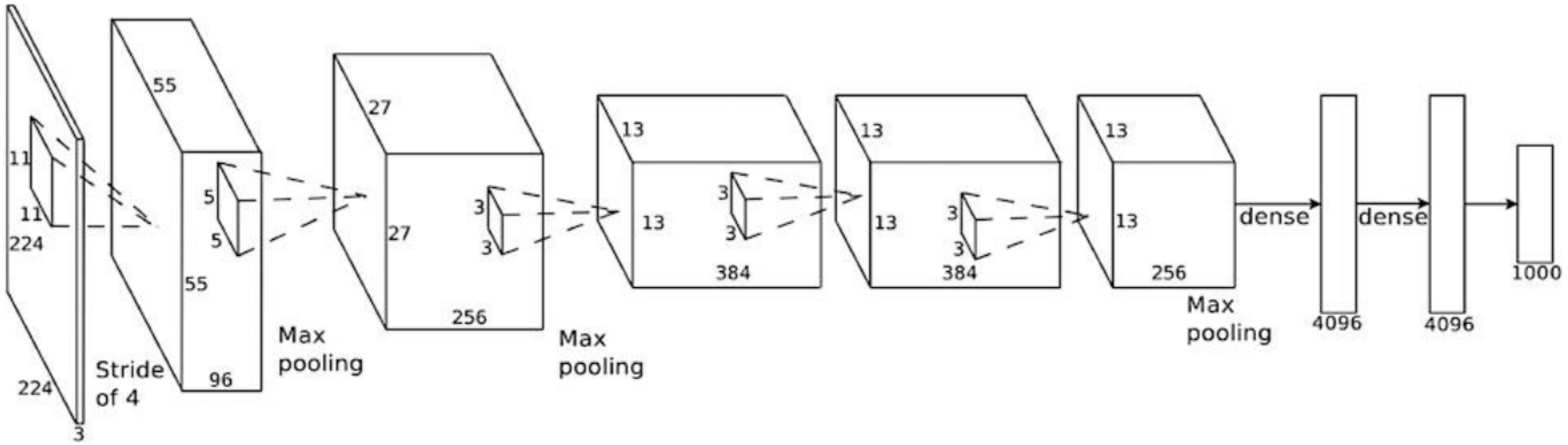


First Strong Results

- AlexNet 2012
  - Winner of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC 2012)
  - Error rate – 15.4% (the next best entry was at 26.2%)



AlexNet: Network Size

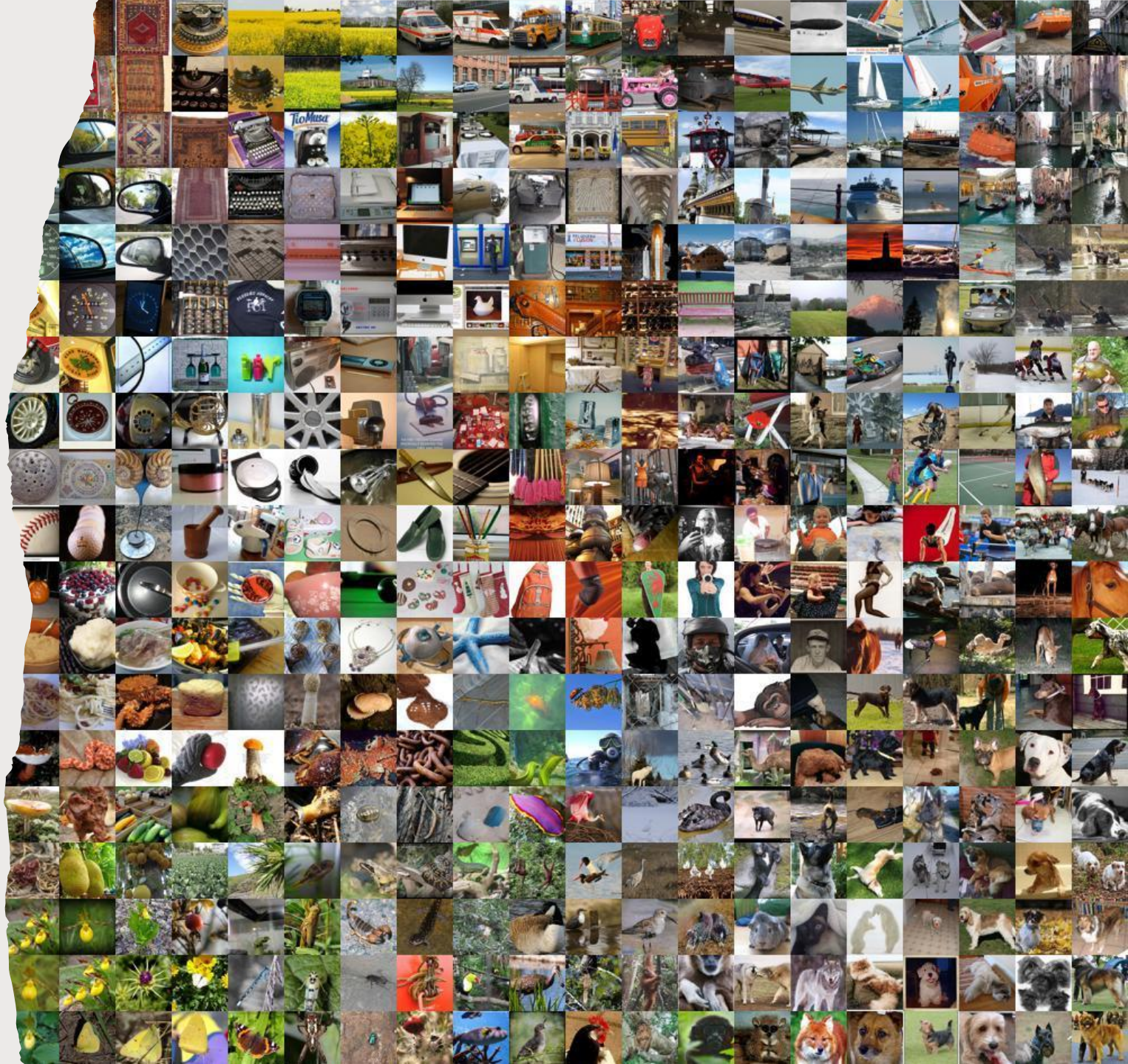


- Input 227x227x3
- 5 convolution layers
- 3 dense layers
- Output 1000-D vector

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8

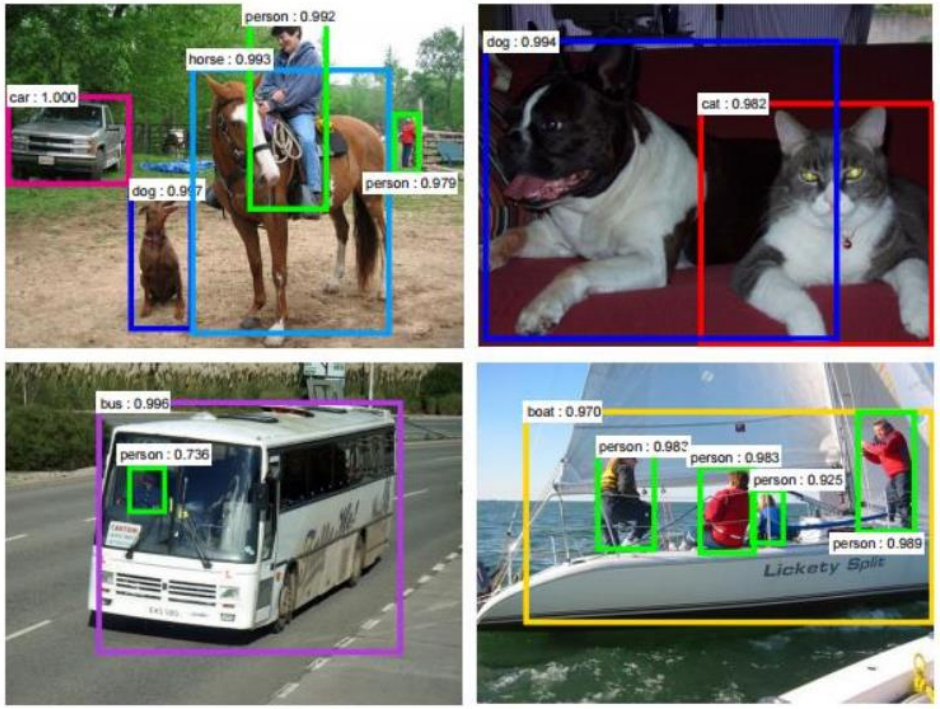
Image-Net

- Almost 14 million Images hand annotated
- Used in various computer vision tasks like classification, object detection etc



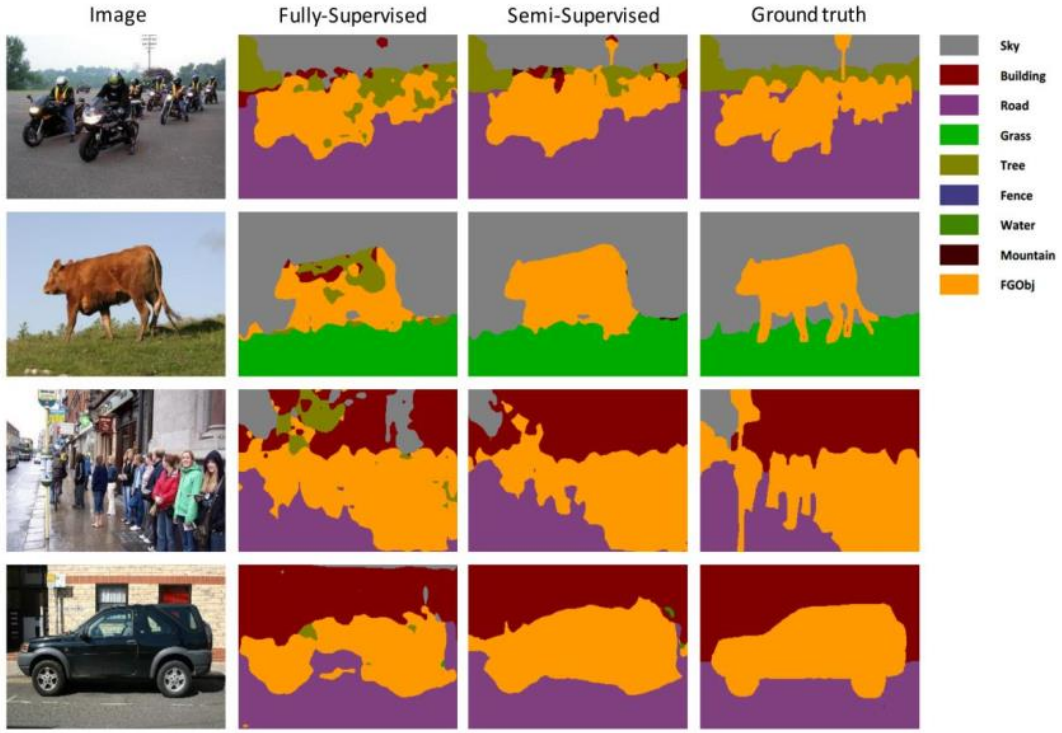
Today: CNNs are everywhere

Object detection



Faster R-CNN: Ren, He, Girshick, Sun 2015

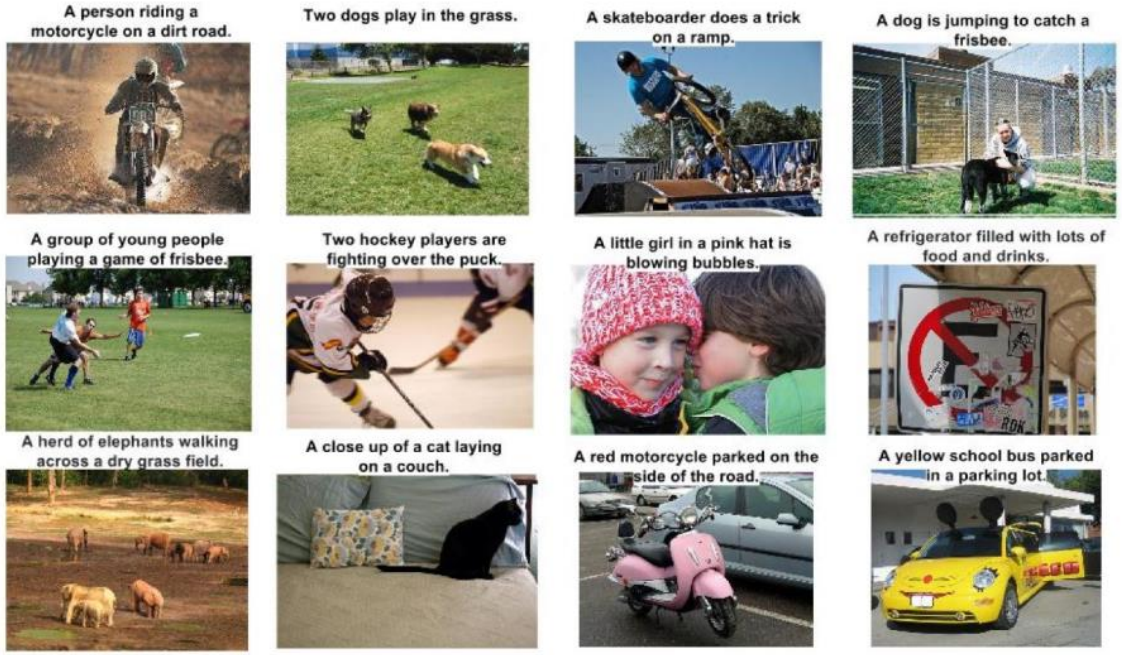
Semantic Segmentation



Semantic Segmentation Using GAN, Nasim, Concetto, and Mubarak, 2017.

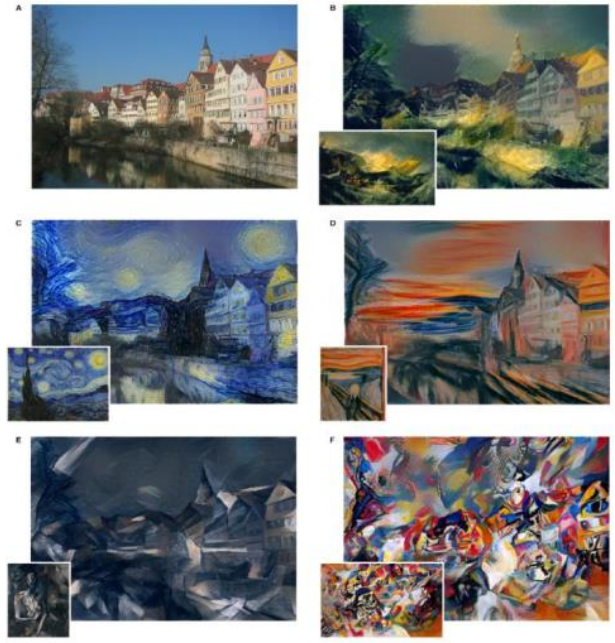
Today: CNNs are everywhere

Image captioning



"Show and tell: A neural image caption generator."
Vinyals, Oriol, et al. CVPR 2015.

Style transfer



A Neural Algorithm of Artistic Style
L. Gatys et al. 2015).

CNN-Not just Images

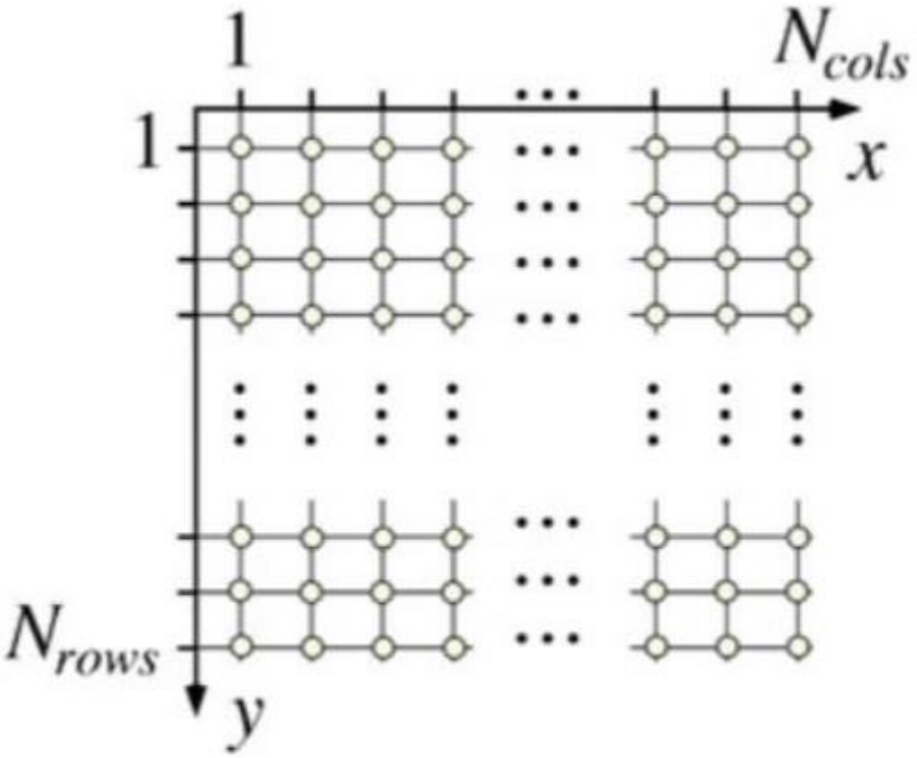
Can be used in NLP:

- Text Classification

Audio Research:

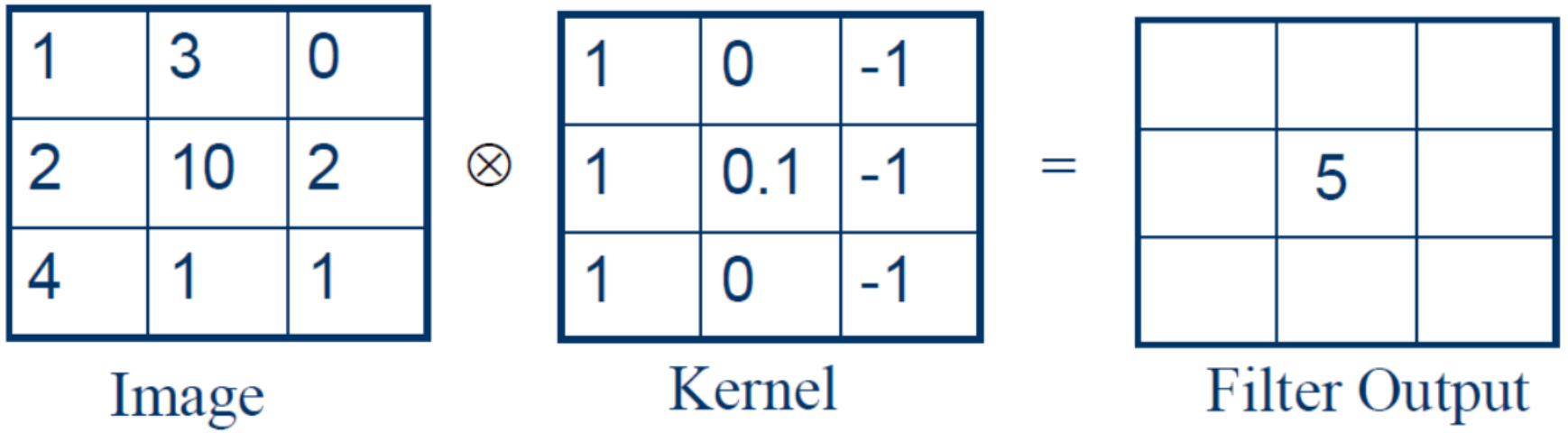
- Speech Recognition

What is a Digital Image?



Filtering

- Output is linear combination of the neighborhood pixels



Two Major Operations in CNN

Convolutions:

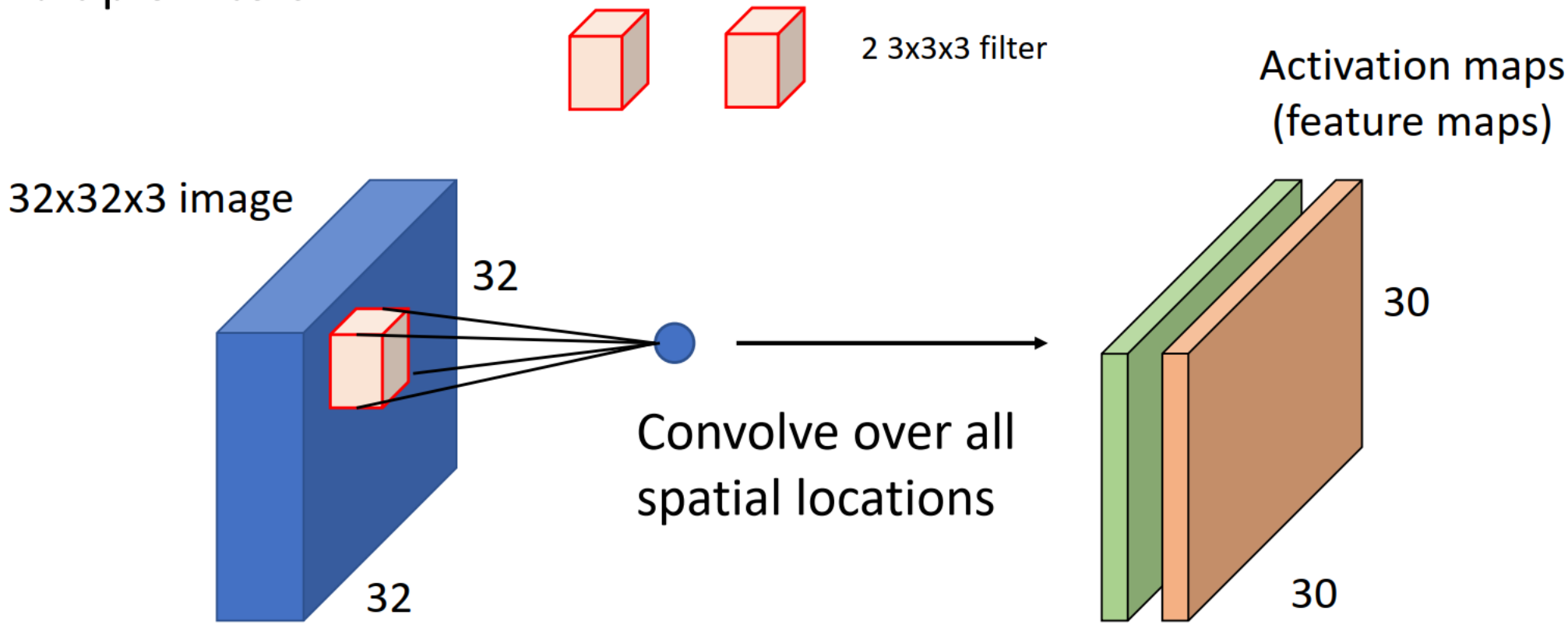
- Convolution is the operation where a filter (also called a kernel) is applied to an input (usually an image or a feature map) to produce a feature map.
- To extract features such as edges, corners, or patterns from the image

Pooling:

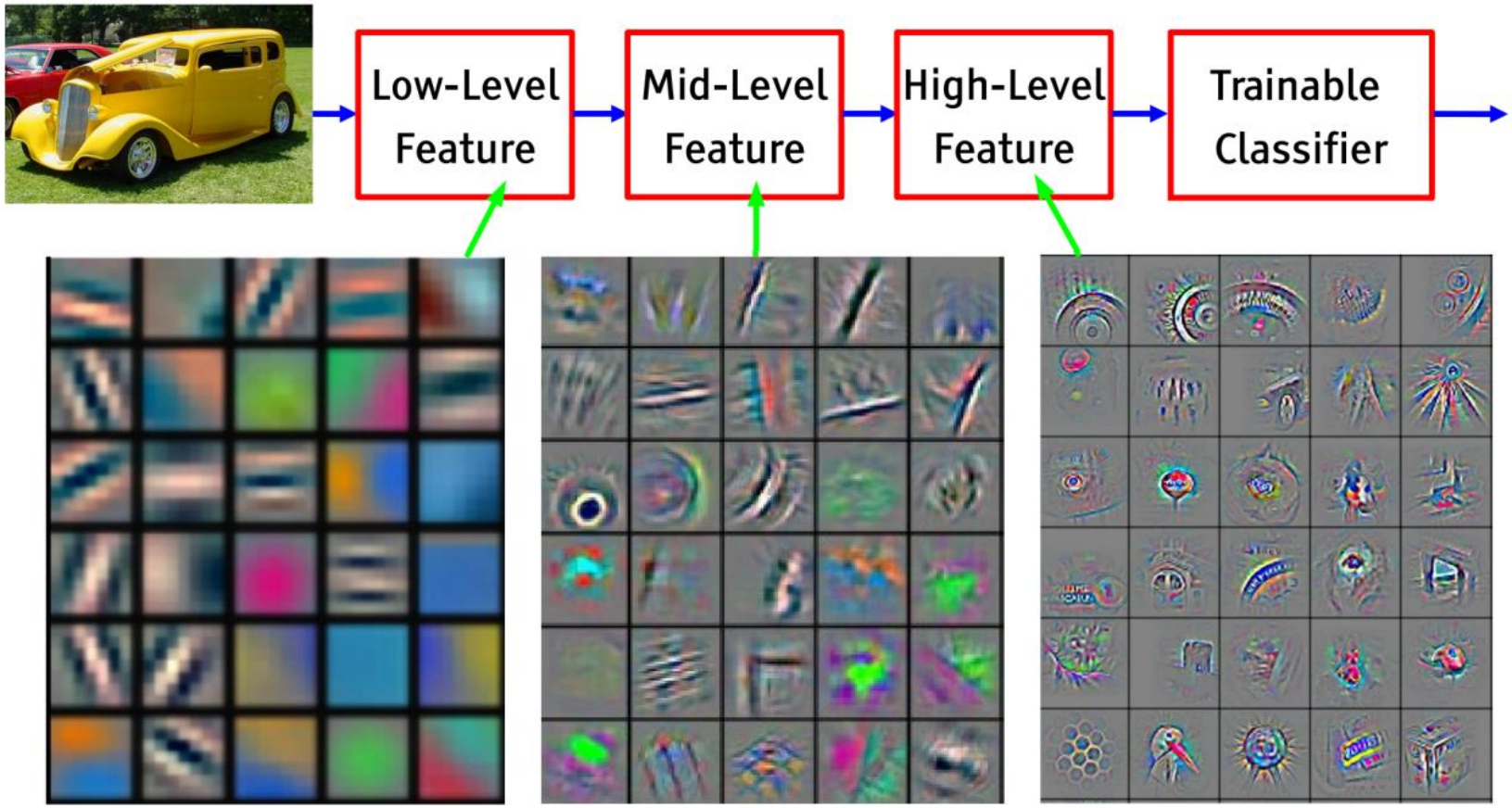
- Pooling is a down sampling operation that reduces the spatial dimensions of the input while preserving important information.
- It is typically used to reduce the computational load and control overfitting by simplifying the representation.

Convolution

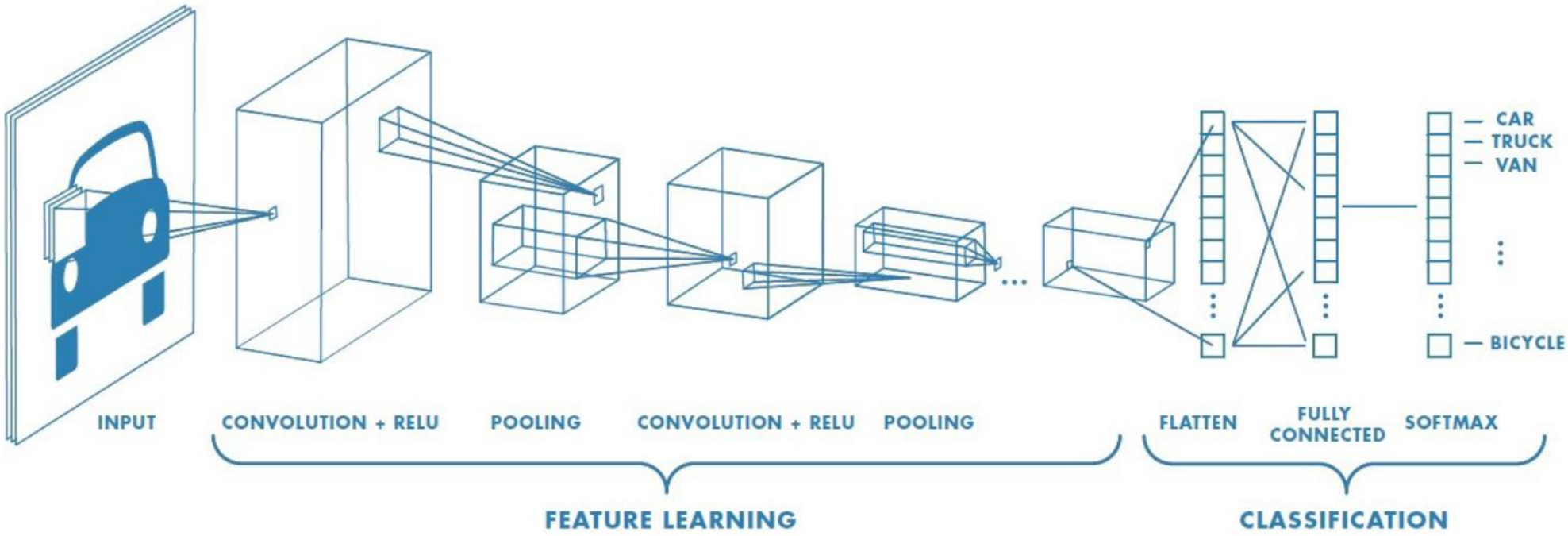
- Multiple filters



Visualizing Convolutions



Summary


CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION

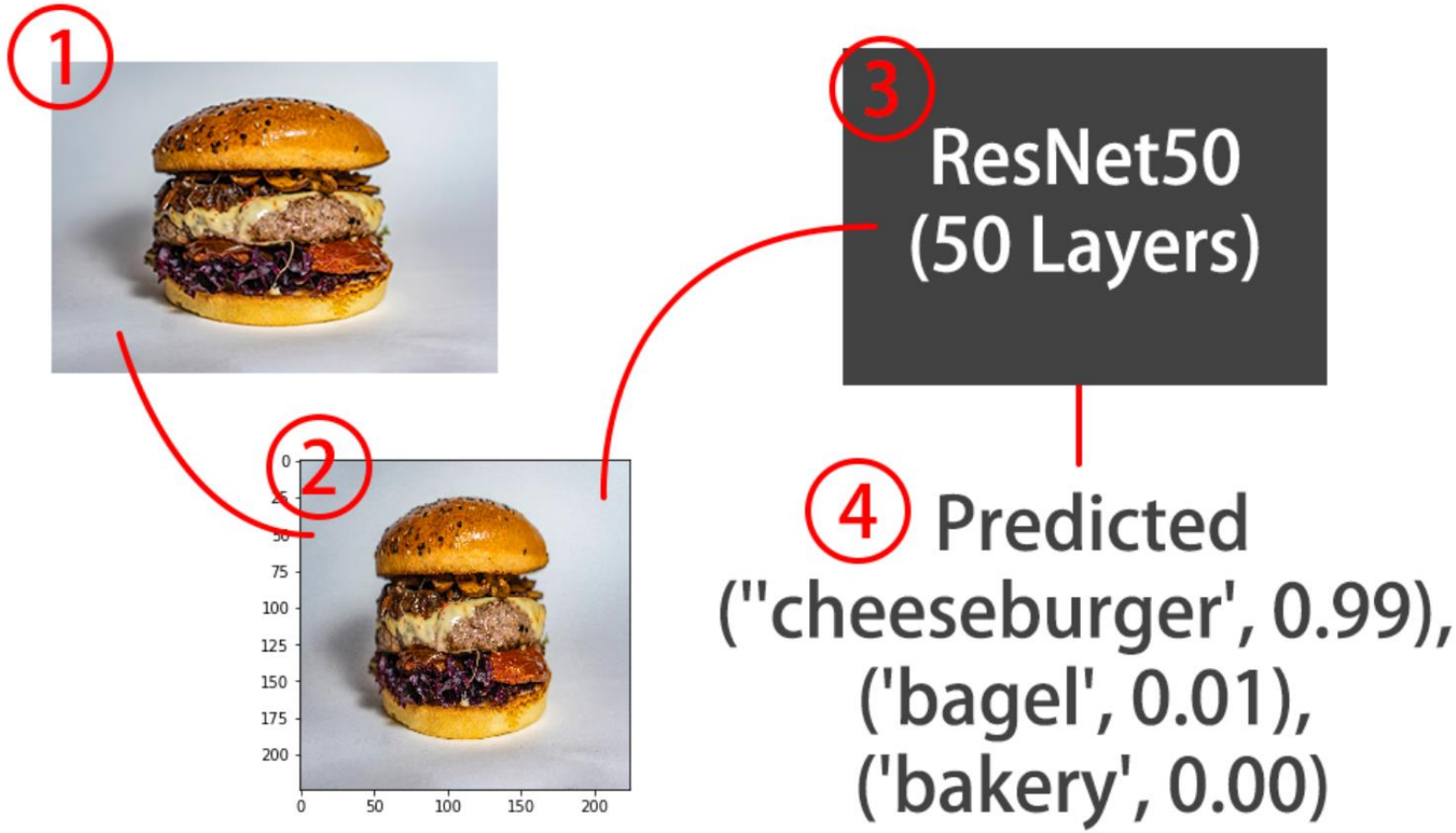
Code Implementation

```
# Import Conv2D layer and Flatten from tensorflow keras layers
from tensorflow.keras.layers import Dense, Conv2D, Flatten
# Instantiate your model as usual
model = Sequential()

# Add a convolutional layer with 32 filters of size 3x3
model.add(Conv2D(filters=32,
                 kernel_size=3,
                 input_shape=(28, 28, 1),
                 activation='relu'))
# Add another convolutional layer
model.add(Conv2D(8, kernel_size=3, activation='relu'))

# Flatten the output of the previous layer
model.add(Flatten())
# End this multiclass model with 3 outputs and softmax
model.add(Dense(3, activation='softmax'))
```

Transfer Learning



Pre-processing Image for ResNet50

```
# Import image from keras preprocessing
from tensorflow.keras.preprocessing import image
# Import preprocess_input from tensorflow keras applications resnet50
from tensorflow.keras.applications.resnet50 import preprocess_input
# Load the image with the right target size for your model
img = image.load_img(img_path, target_size=(224, 224))
# Turn it into an array
img = image.img_to_array(img)
# Expand the dimensions so that it's understood by our network:
# img.shape turns from (224, 224, 3) into (1, 224, 224, 3)
img = np.expand_dims(img, axis=0)
# Pre-process the img in the same way training images were
img = preprocess_input(img)
```

Using ResNet50 Model

```
# Import ResNet50 and decode_predictions from tensorflow.keras.applications.resnet50
from tensorflow.keras.applications.resnet50 import ResNet50, decode_predictions
# Instantiate a ResNet50 model with imagenet weights
model = ResNet50(weights='imagenet')
# Predict with ResNet50 on our img
preds = model.predict(img)
# Decode predictions and print it
print('Predicted:', decode_predictions(preds, top=1)[0])
```

```
Predicted: [('n07697313', 'cheeseburger', 0.9868016)]
```