"Unveiling the Magic of Convolutional Neural Networks: A Hands-On Journey into Image Recognition"

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Submitted by
KARAN SINGH RAWAT
[RA2111003011002]
KUMAR ASHISH
[RA2111003011067]

Under the Guidance of
Dr. Sridhar S S
Assistant Professor
CTECH

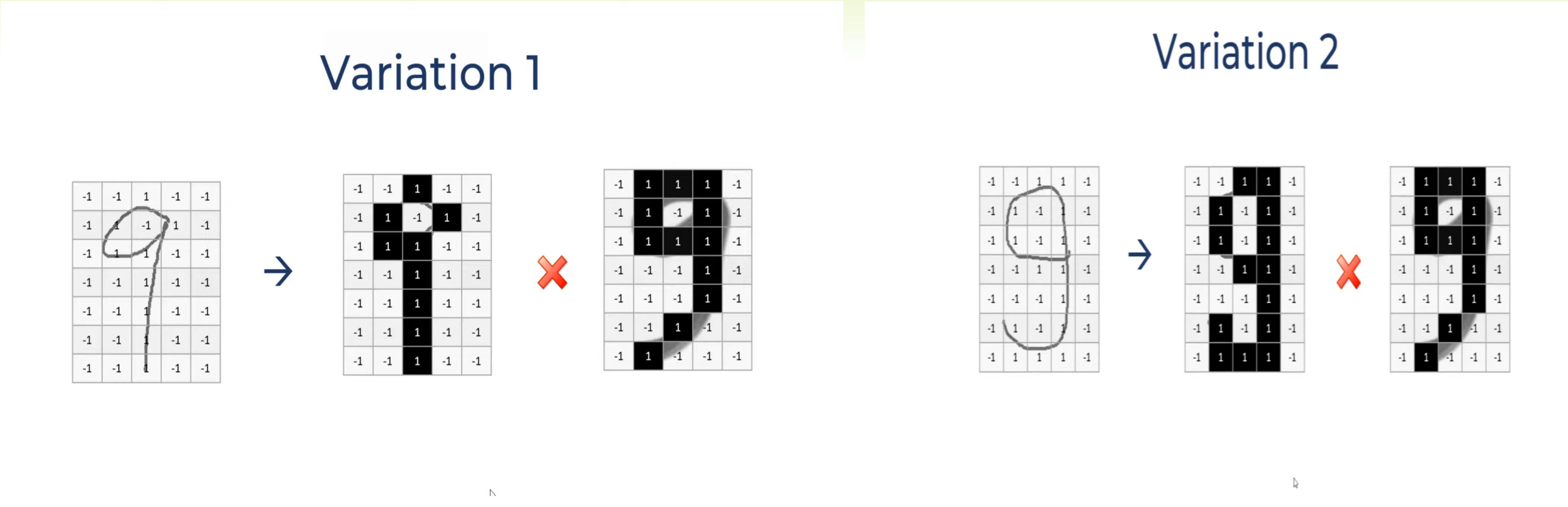
Table of Contents

Introduction

Imagine you're teaching a friend how to recognize handwritten digits. You show them a picture of a messy "9" and explain its key features: the curved line at the top and the straight line coming down.

how does a computer do this? Traditionally, it would need very specific instructions to identify each pixel (tiny building block) in the image. This is like giving your friend a giant grid with instructions to check for a specific pattern of black and white squares everywhere.

The problem? Handwritten digits can be messy! A slight slant or shift can throw off the whole grid system. The computer wouldn't recognize the "9" because the exact pattern it was looking for wouldn't be there.



Here's where CNNs come in!

Instead of a giant grid, CNNs use small filters that slide across the image, looking for specific patterns like curves, lines, and corners.

If Computer use ANN model (Grid/block) for large Image it’s Not Suitable - That nearly 24 million hidden layers!

Image size = 1920 x 1080 X 3

First layer neurons = 1920 x 1080 X 3 ~ 6 million

Hidden layer neurons = Let’s say you keep it ~ 4 million

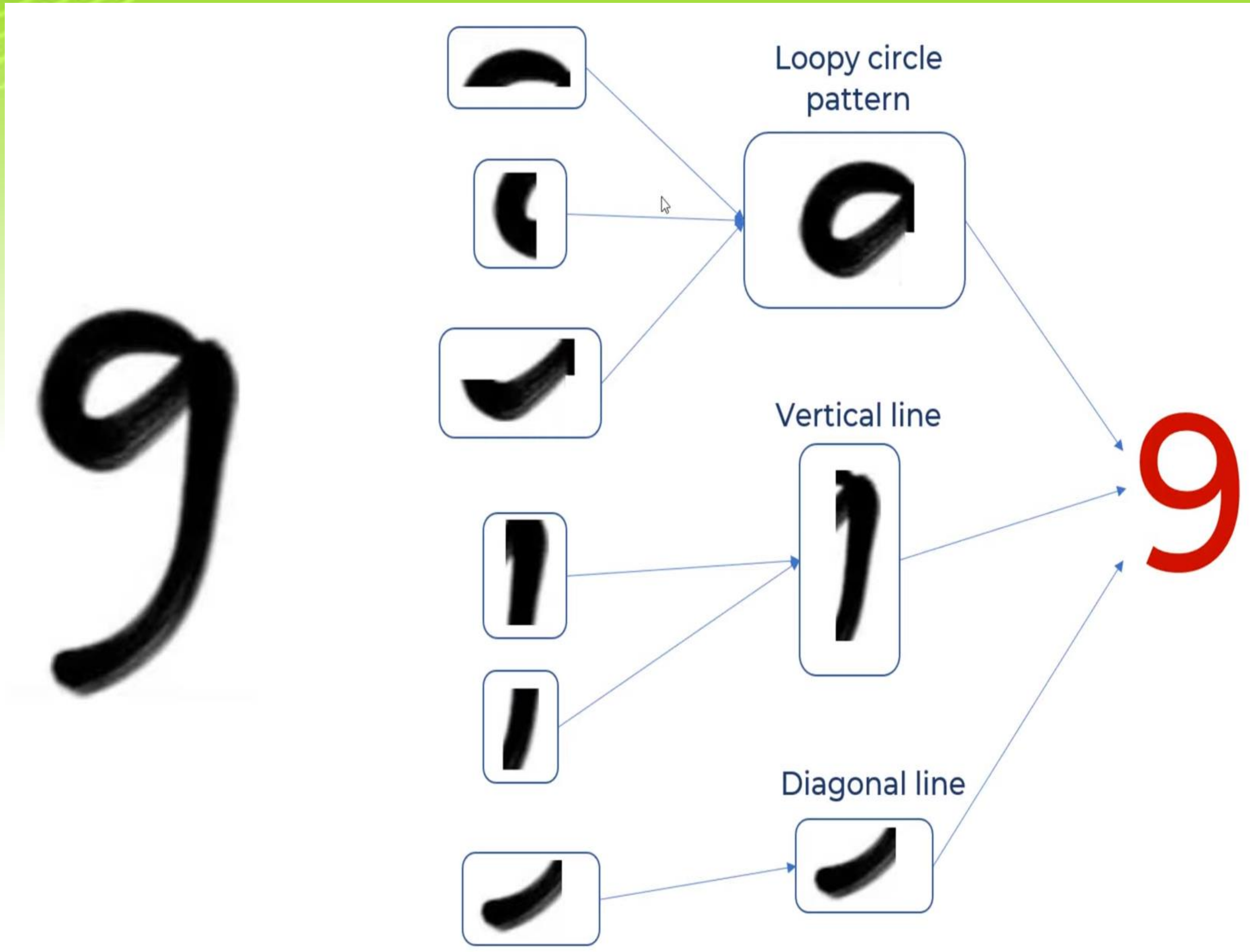
Weights between input and hidden layer = 6 mil * 4 mil
= 24 million

Part II : Understanding The Topic

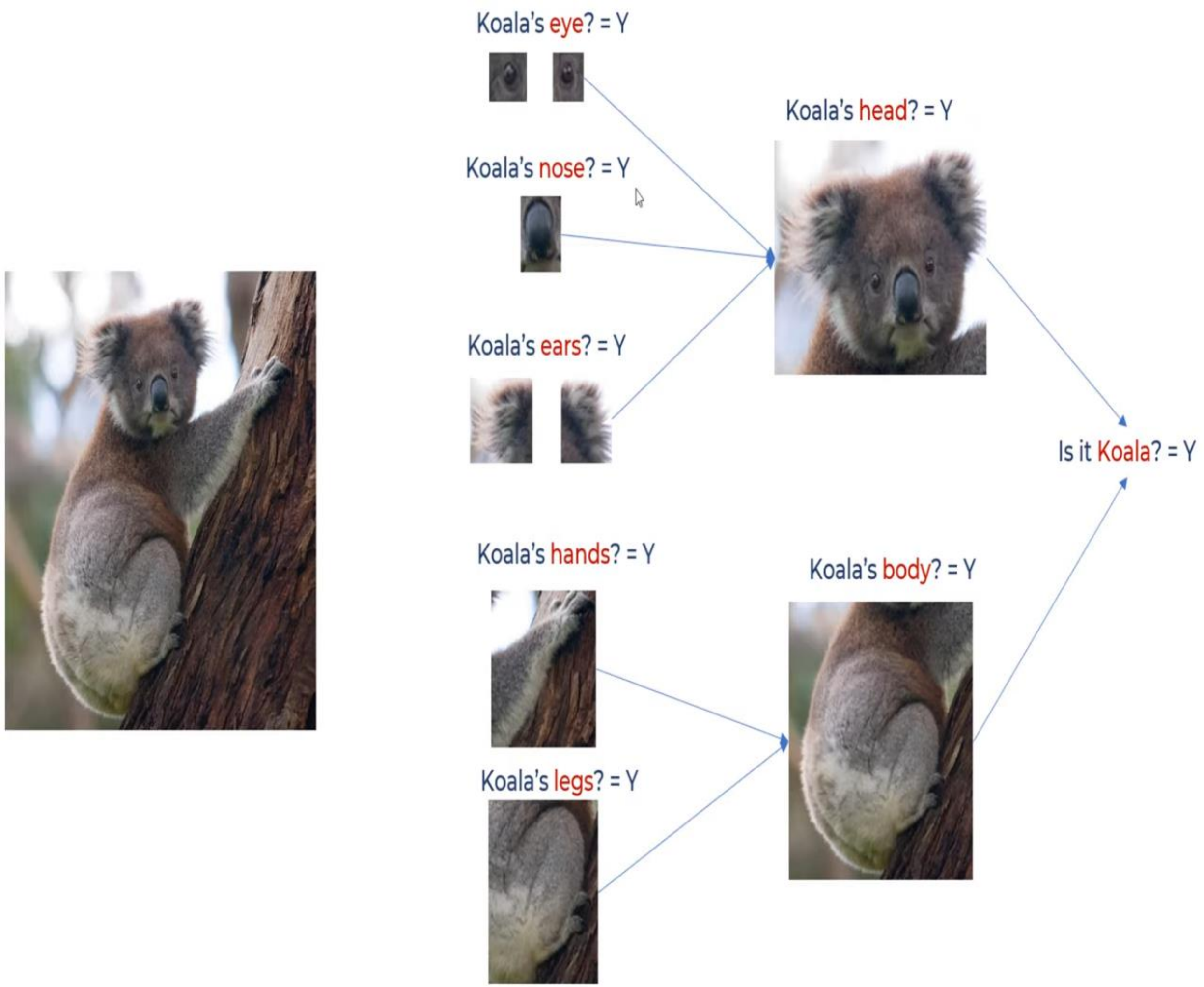
Disadvantages of using ANN for image classification

1. Too much computation
2. Treats local pixels same as pixels far apart
3. Sensitive to location of an object in an image

How CNN Work ?

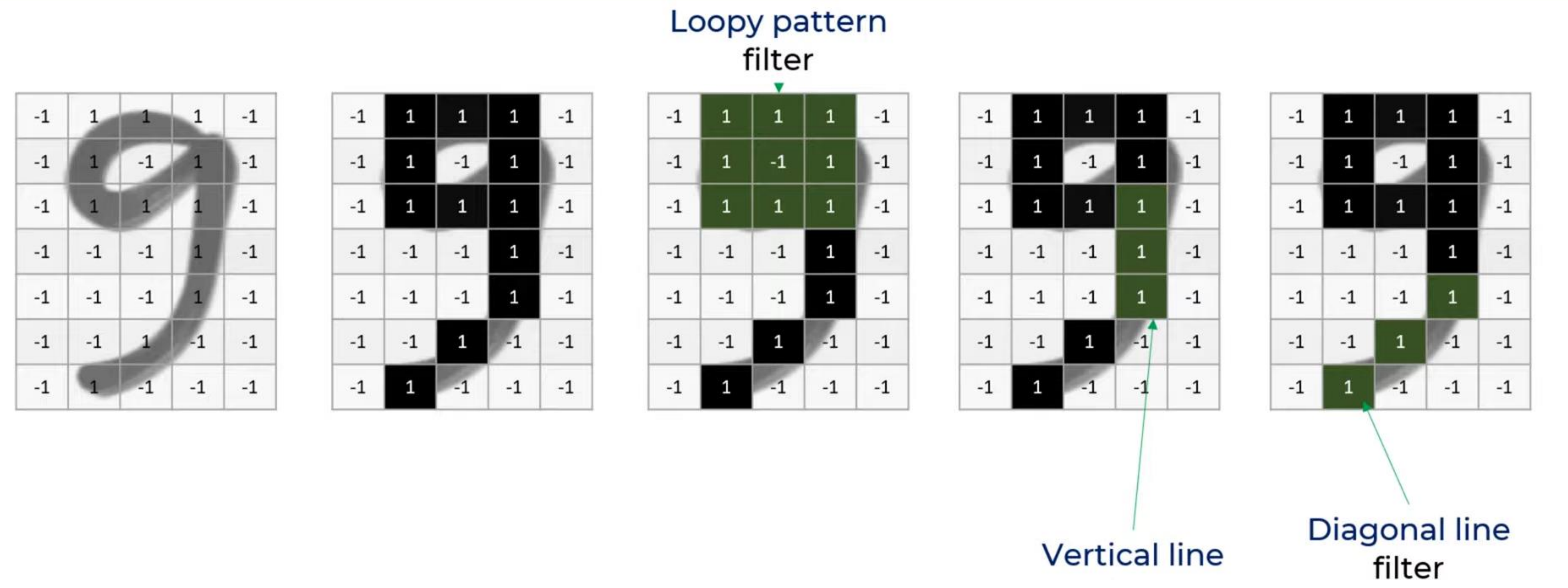


Loopy circle pattern
Vertical line
Diagonal line
9



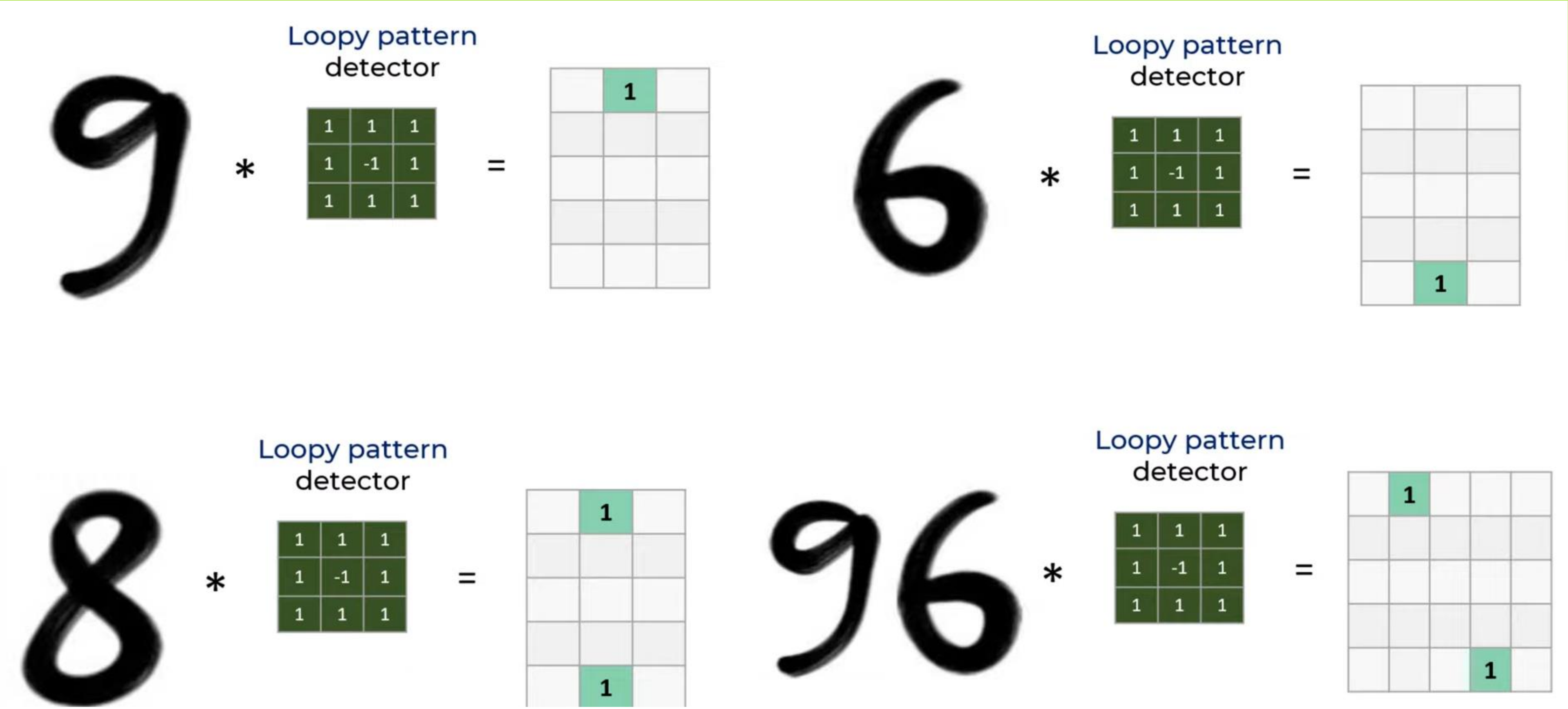
Koala's eye? = Y
Koala's nose? = Y
Koala's ears? = Y
Koala's head? = Y
Koala's hands? = Y
Koala's legs? = Y
Koala's body? = Y
Is it Koala? = Y

- **we've applied different filters (head, middle, tail) to create three separate feature maps for the digit nine.**
- **We can repeat this process for other objects like a koala, using filters for eyes, nose, and ears, resulting in three distinct feature maps.**
- **However, these feature maps are like individual clues – we need to combine them for a final decision.**

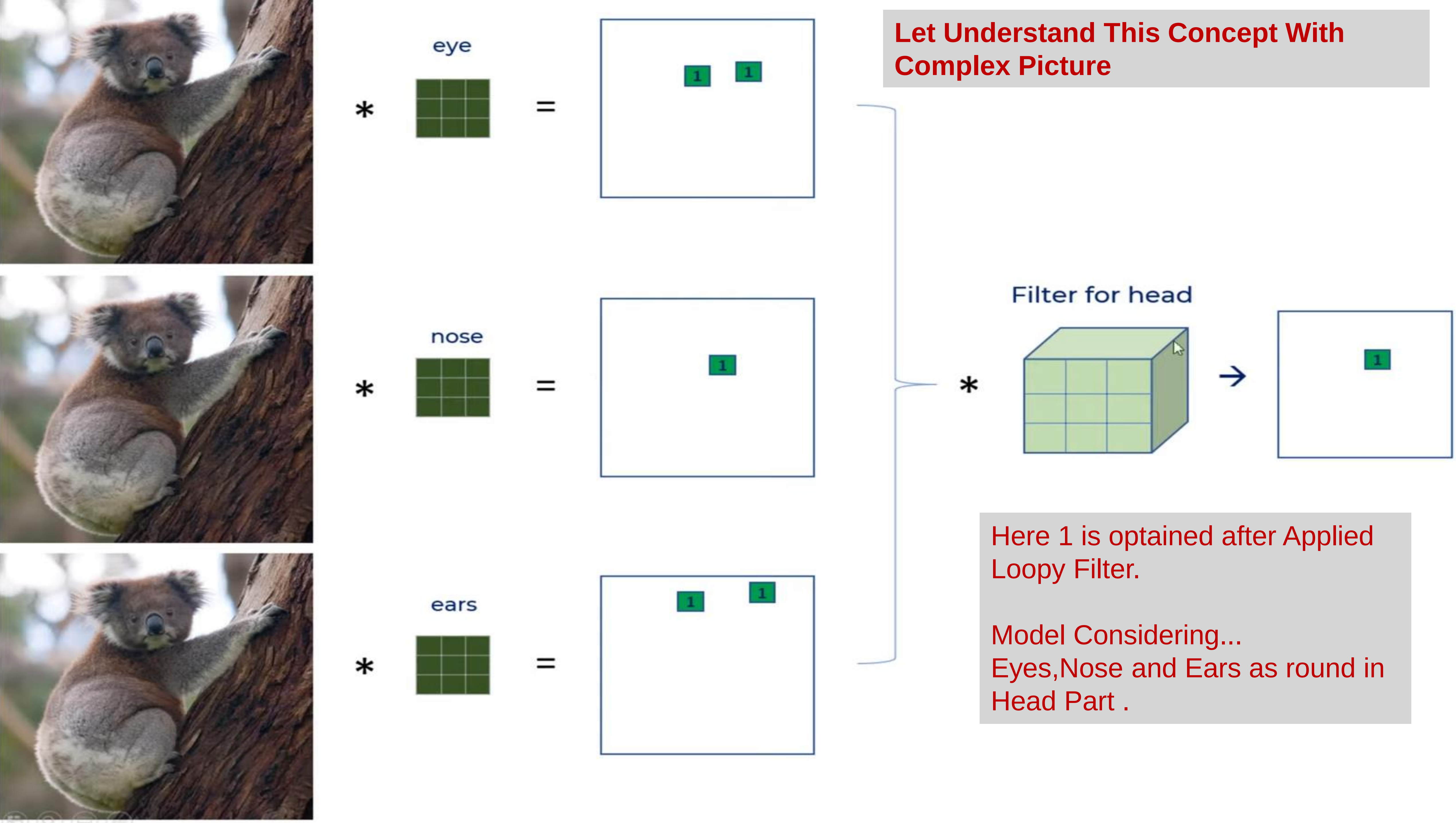


Filters are nothing the Feature’s Detectors

Applying Loopy Pattern Filter



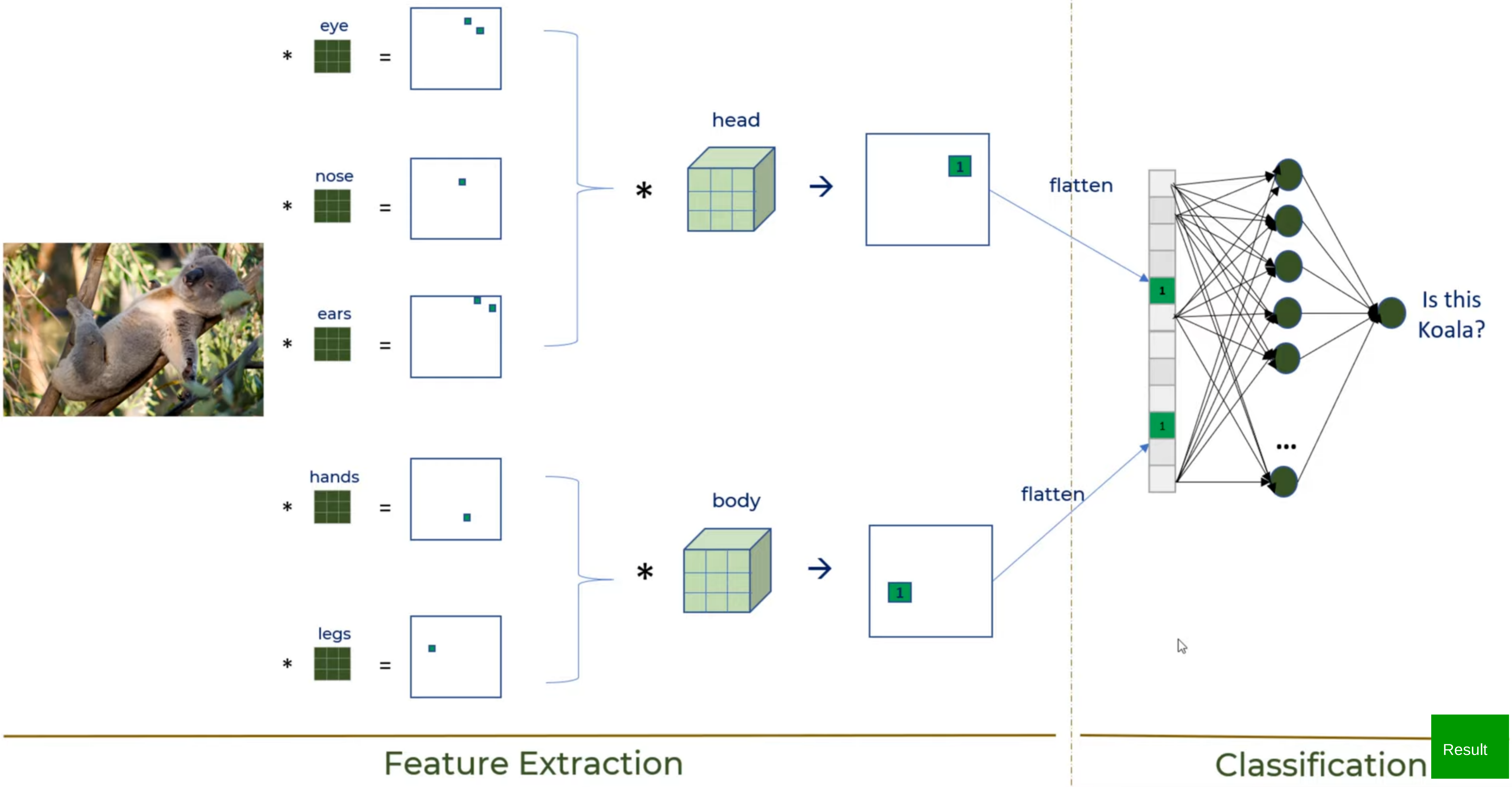
Let Understand This Concept With Complex Picture



Here 1 is optained after Applied Loopy Filter.

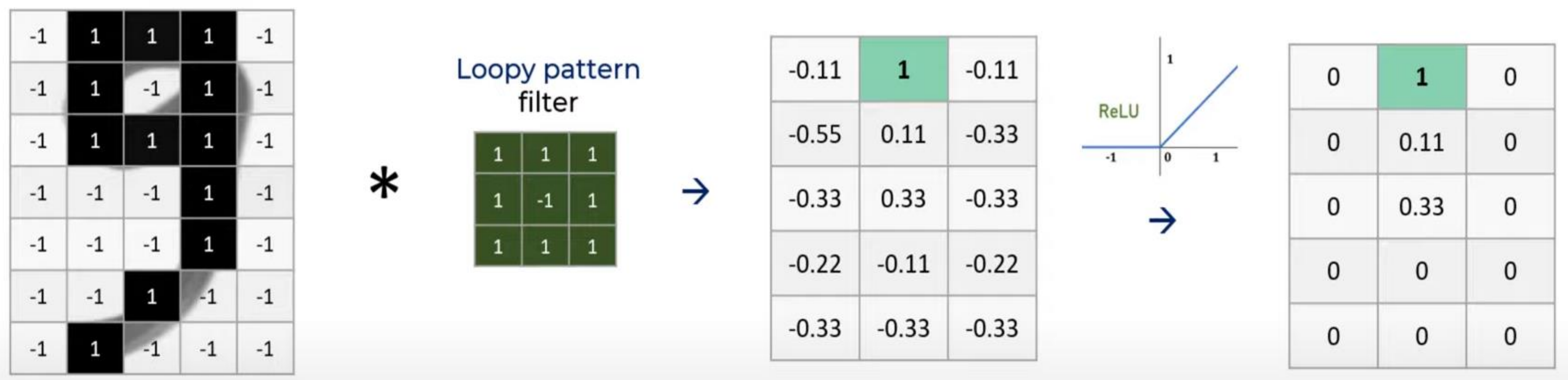
Model Considering...
Eyes,Nose and Ears as round in Head Part .

Topic III : Analyze The Method



ReLu Function, helps With Making The Model NonLinear

Its convert the Negative values to : 0
while positive remain’s Same



Pooling Layer is used to Reduce the Size

0	1	0
0	0.11	0
0	0.33	0
0	0	0
0	0	0

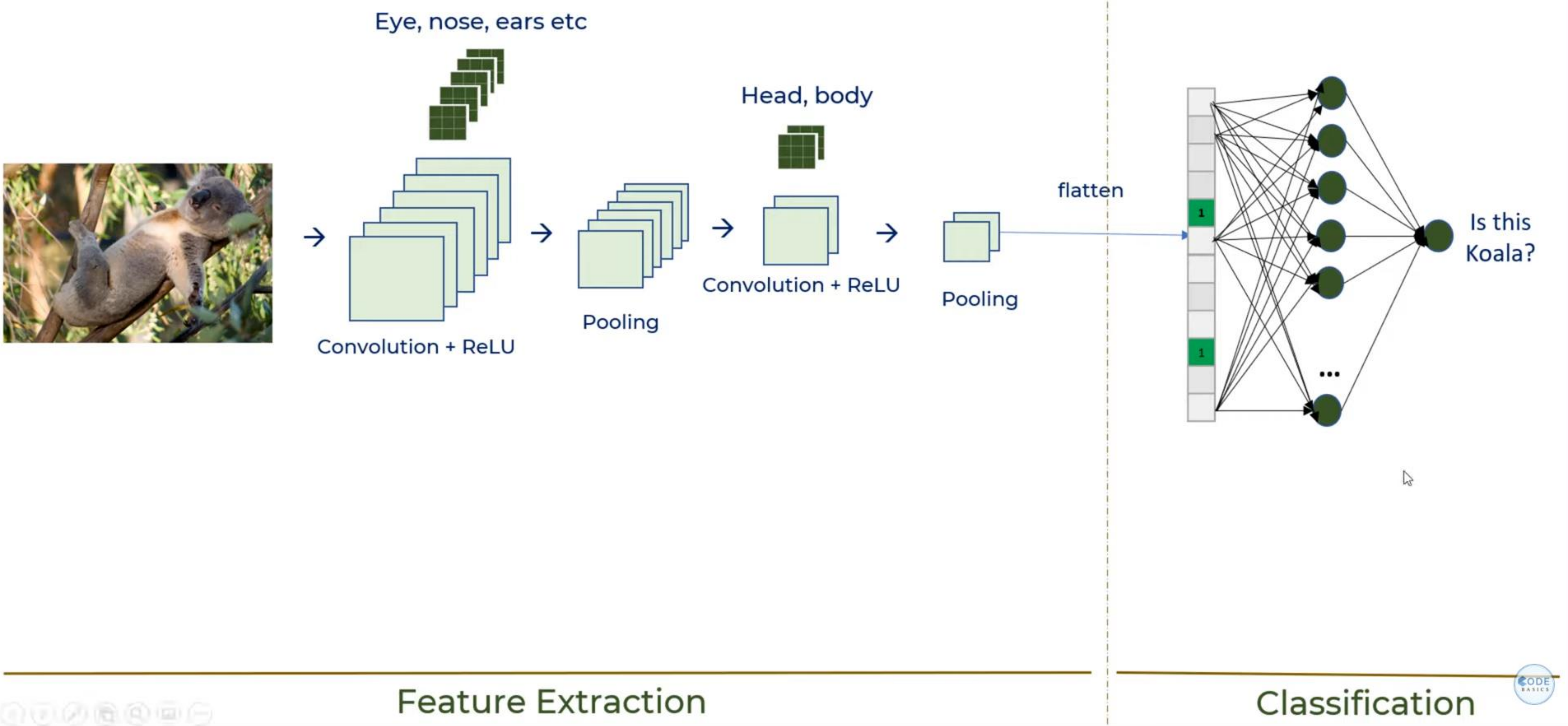
Here It Took Pair of 4 Points / pixcels

- **In first Four 1 is max then Stride slide by 1**
- **In next Four points 1 is Max so it would be picked**
- **Similary for whole Image it applied and Decrease the Size of image**

1	1

2 by 2 filter with stride = 1

So This is How Complete Convolutional Neural Network Look LIke...
Feature Extraction + Classification



Convolution

- Connections sparsity reduces overfitting
- Conv + Pooling gives location invariant feature detection
- Parameter sharing

ReLU

- Introduces nonlinearity
- Speeds up training, faster to compute

Pooling

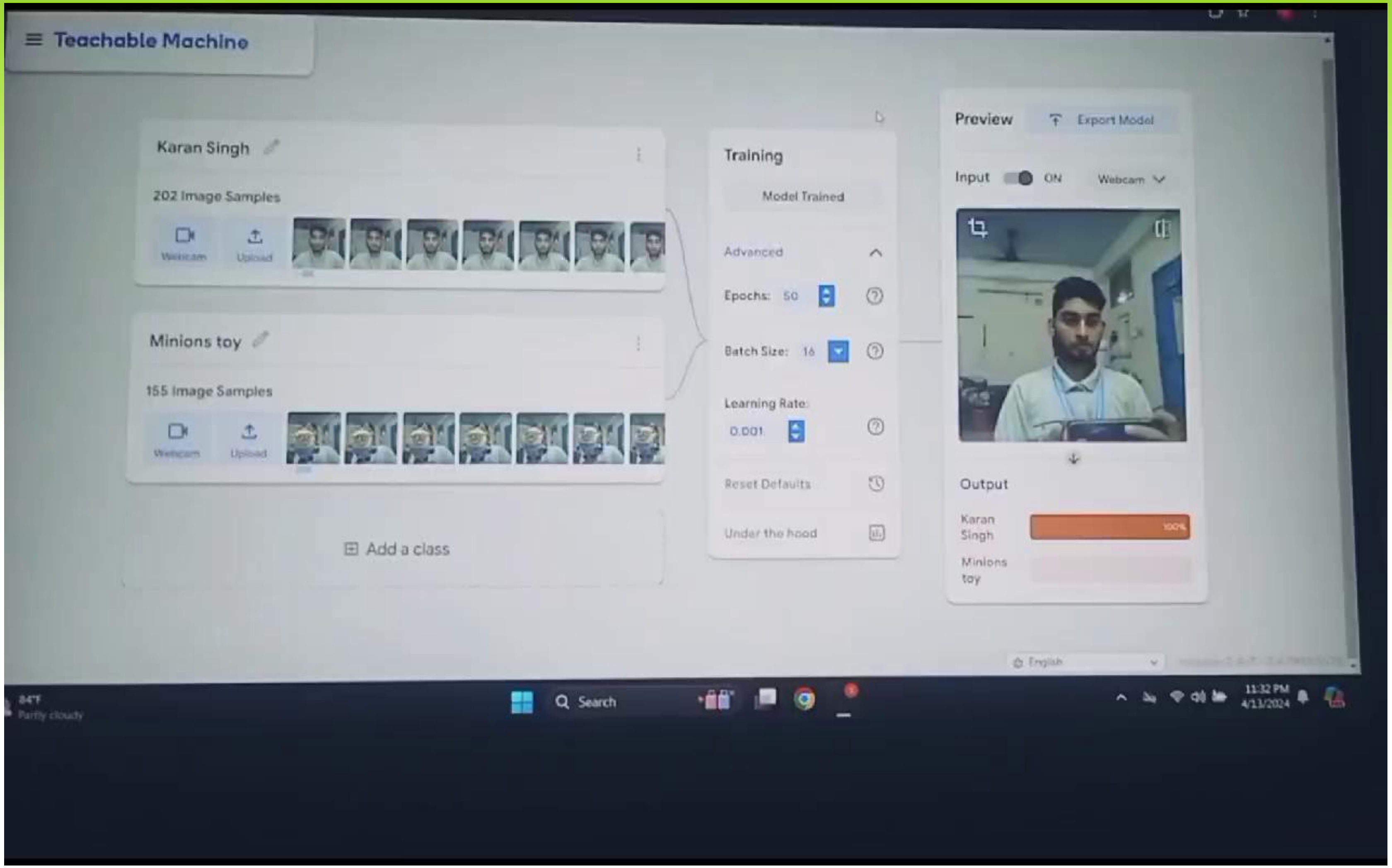
- Reduces dimensions and computation
- Reduces overfitting
- Makes the model tolerant towards small distortion and variations

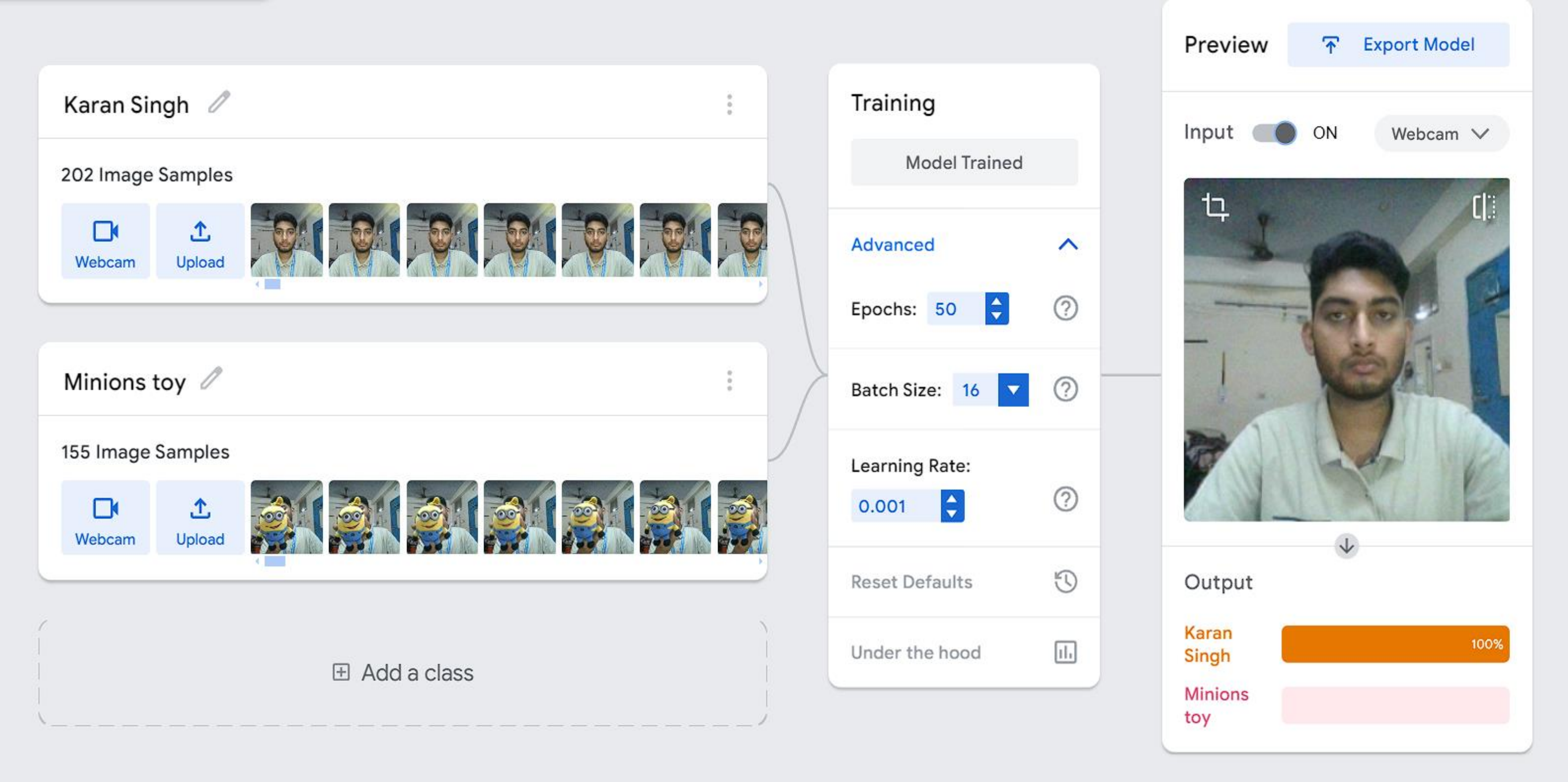
Topic IV : Intereactive Learning Showcase

- **CNN Model after Trained with few Samples Able to Differtiate between me and toys.**

- **Capable enough to detect even moving Object with Different Angle’s.**

Click on the Video to Play →





202 sample Image of Human Being , 155 picture of Toys , Epochs -50 , Where as Learning Rate is -0.001

Part V : Conclusion

In summary, convolutional neural networks (CNNs) offer a groundbreaking approach to image recognition tasks by efficiently extracting features, reducing computational complexity, and achieving robust performance across diverse datasets. Their ability to handle variations in image presentation while mimicking human visual recognition capabilities makes them invaluable tools in the field of artificial intelligence.

Regular computers struggle with image variations. Convolutional Neural Networks (CNNs) solve this by automatically learning tiny features (eyes, curves) from images. This allows CNNs to recognize objects despite variations, but requires a lot of computation. That's why CNNs are powerful but complex!.

References

- GCR -SRM PPT’s
- S.N.Sivanandan, S.N.Deepa, Principles of Soft Computing, 3 edition , Wiley Publications
- Visualizing CNNs and their Filters [Blog] (Distill.pub): https://distill.pub/

  Deeplearning.ai - Convolutional Neural Networks [Website]: https://www.coursera.org/learn/convolutional-neural-networks

  https://cs231n.stanford.edu/2021/slides/2021/lecture_5.pdf

  https://youtu.be/zfiSAzpy9NM?si=oDumUzPityniQWg-

Source Code

```
from keras.models import load_model  # TensorFlow is required for Keras to work
from PIL import Image, ImageOps  # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_Model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open("<IMAGE_PATH>").convert("RGB")

# resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)
```

Model, Source Code and Presentation Uploded on GCR and Github.

Github link - https://github.com/KumarAshish19/ImageRecogniton-Convolutional-Neural-Network-

or

https://github.com/kk8873/ImageRecogniton-Convolutional-Neural-Network-

Thankyou