Final Project

ECE5268: Theory of Neural Networks,

Spring 2023

Finding My Pet – My home buddy

Dhruthi Sridhar Murthy – 904019545

Instructor: Dr. Anagnostopoulos, Georgios C.

Contents

Convolution Neural Networks

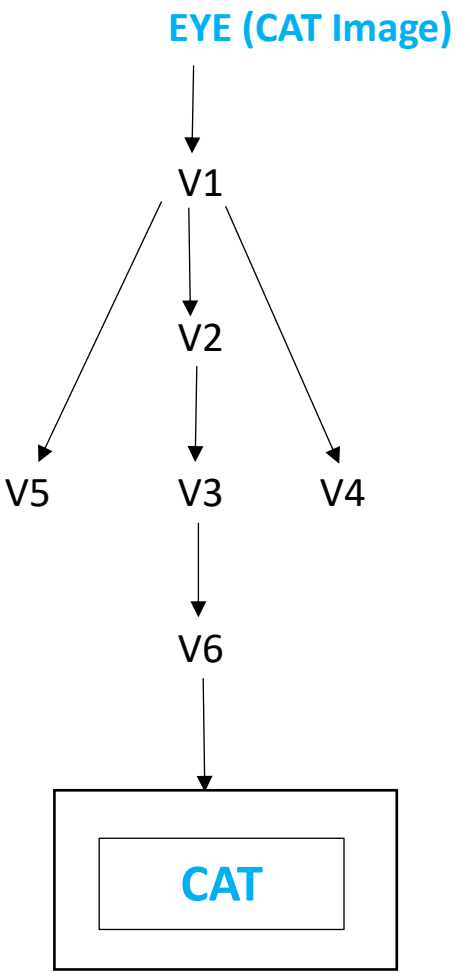
Introduction

- What is a Convolution Neural Network?

The features of the input training samples are extracted using the convolution layers. There are various layers or a set of filters for the feature extractions. Convolution neural networks are compatible with image recognition and processing tasks. The higher the coating, the feature extraction is elevated, i.e. Initially, the first layer of convolution captures simple features in layer one and is less complex compared to the features caught in the last layer.

Convolution is the critical component of a CNN, where the filters are applied to the input image to extract features such as edges, textures, and shapes. The convolution neural network is like the human brain. As we know, the cerebral cortex is located at the backside of our head; inside this, there is a visual cortex responsible for seeing the image. Let's try to understand the concept well. Now when we humans watch an image, say a cat, how do we recognize that it's a cat? So, the image will go through the sensory organ, our eyes, and then it will reach the visual cortex layer. These have various layers, as indicated below. So, each layer will perform new tasks to get the image clarity or details well. The information is transferred from one layer to another; then,

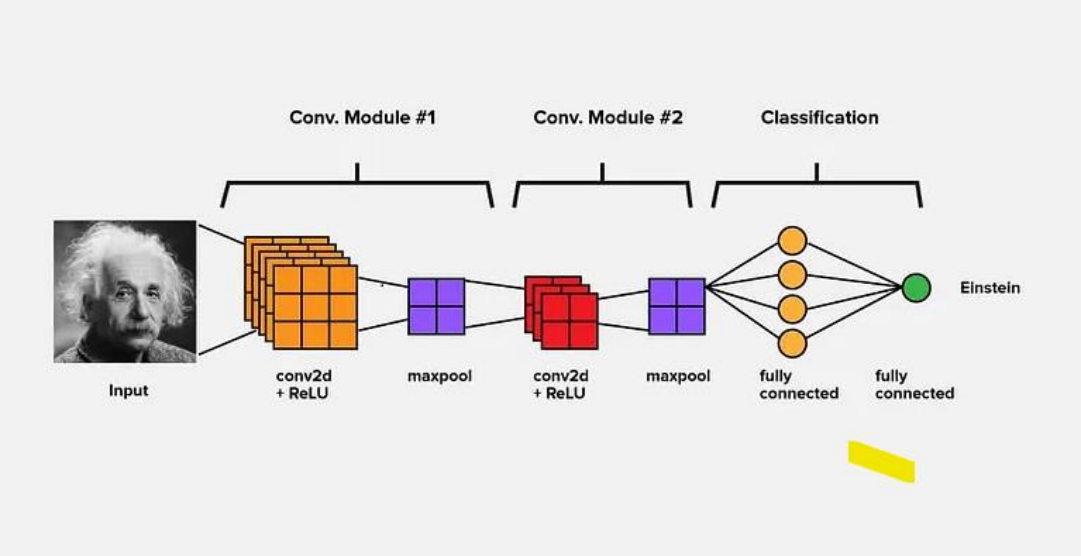
finally, the object is detected by the sensory organ, stating it's a cat. The process indicated in the image is the convolution of neural networks.



The convolution layers are the critical component of the CNN, where filters are applied to the input images to extract features such as edges, textures, and shapes.

The output of the CNN is passed through an activation unit called ReLU (Rectified Linear Unit). This unit converts the data into its non-linear

form. Then the output is passed through the pooling layers, which are used to down-sample the feature maps. CNN is trained using the large dataset of labeled images, where the network learns to recognize patterns and features associated with specific objects or classes. The central concept of pooling is to assume that the values of the image pixels are nearly similar.



The above image describes the various steps involved in the image processing of convolution neural networks.

Further information about CNN can be found at:

Neural Networks

Overview of CNN Layers

Ref: CNN

Convolution: Features from the input data samples are extracted.

ReLU: Data are converted to a non-linear form.

Pooling: The layer aims at step by step cut down of the dimensionality of the data. The redundancy of the features is removed.

Figure 3: Max-Pooling Operation

Fully connected layer: Flatten the high-level features learned by convolutional layers and blend all the elements.

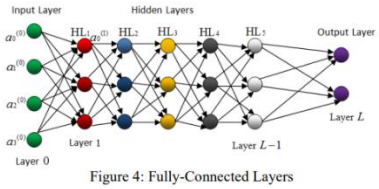


Figure 4: Fully-Connected Layers

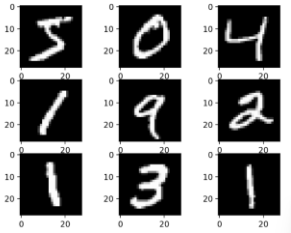
Dropout and regularization: Make the model robust to noise

Multiple Convolution Layer: Extract high-level or more complex features.

Extraction of feature – MNIST

The MNIST dataset is handwritten, a standard computer vision and deep learning dataset.

It is a dataset of 60,000 small square 28 x 28-pixel grayscale images of handwritten single digits between 0 and 9. The task is to classify a given picture of a handwritten number into one of 10 classes representing integer values from 0 to 9 inclusively.



This is an example of the MNIST dataset – showing the natural handwritten nature of the image to be classified.

CIFAR Dataset

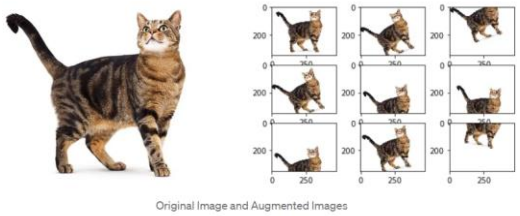
The CIFAR dataset is the collection of images commonly used to train machine learning and computer vision algorithms. It is one of the widely used datasets for machine learning research. The dataset consists of 60,000 32x32 color images in 10 different classes. The ten classes represent airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. There are 6000 images of each class.

In this classification, we are extracting our cat and dog to understand where these pets belong to? Whether the pet owner owes it a dog or a cat. This is what we are trying to analyze with this dataset.

Custom Image Augmentation

Image Augmentation

This is basically used to expand the dataset artificially. Image augmentation artificially creates training images through different ways of processing or a combination of multiple processing, such as random rotation, shifts, shear and flips, etc.



Original Image and Augmented Images

The above image indicates the image augmentation of the image of the cat for various processing. Image Augmentation samples are generated using image to increase of existing data sample set by nearly 3x to 4x times. In keras, we attain image augmentation with the help of the function called as ImageDataGenerator. The basic outline of the function definition is shown below:

```
datagen = ImageDataGenerator(
        zoom_range=0.2,
        shear_range=0.2,
        rotation_range=40,
        fill_mode='nearest',
        horizontal_flip=True,
        preprocessing_function = image_contrast_adjusment)
```

This is the function to initialize data augmentation Parameters.

Custom Image Augmentation

We changed our Keras dataset to generate API to make the image more powerful.

Histogram Equalization

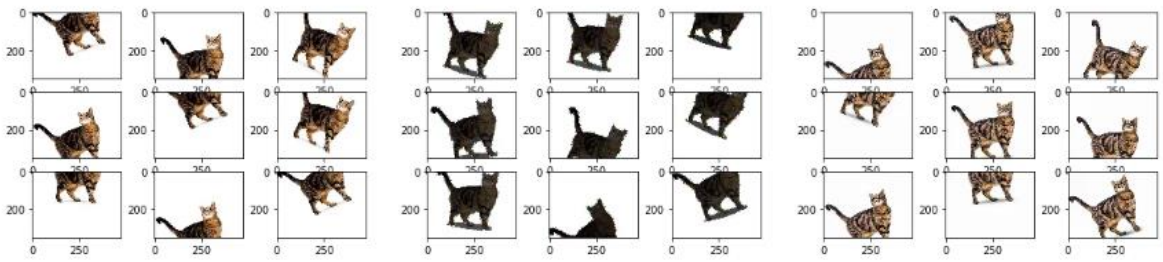
This technique increases the global contrast of an image using the image intensity histogram. The equalized image has a linear cumulative distribution function. The equalized image has a linear cumulative distribution function. This method has no parameters, but it sometimes looks unnatural.

An alternative is the adaptive histogram equalization (AHE) which improves local contrast of an image by computing several histograms corresponding to different sections of an image.

Contrast limited adaptive histogram equalization was developed to prevent the over-amplification of noise resulted from AHE.

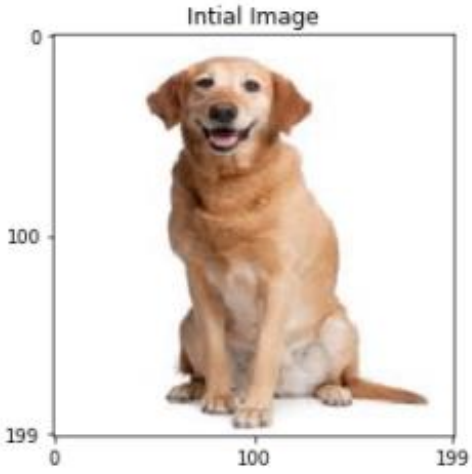
```
def AHE(img):
    img_adapteq = exposure.equalize_adapthist(img, clip_limit=0.03)
    return img_adapteqdatagen =
ImageDataGenerator(rotation_range=30, horizontal_flip=0.5,
preprocessing_function=AHE)
```

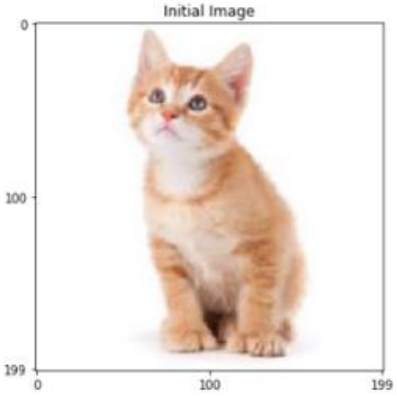
The above code is used to implement the AHE. (Adaptive Histogram Equalization)



Augmented Images using Contrast Stretching (left), Histogram Equalization (middle) and Adaptive Histogram Equalization (right)

Experimental results





These images are classified as a dog and cat which belongs to which house, or we can classify them as a group of pets that belong to the owner.