**ITAI 1378 Computer Vision**

**Patricia McManus**

**Mayela Miguel**

**Image classification with SVM**

**When I initially saw the code, I felt absolutely overwhelmed. The comments above the code explaining what's happening are very helpful. I had never heard of SVM or CIFAR-10 before. I do admit it made me anxious about how I was going to approach trying to understand this assignment, but as I was reading through and taking notes, it became easier and easier to understand what was going on, and it gave me confidence for next future assignments to not undermine my discipline.**

**The hardest concept that I had a little bit of trouble with was flattening the images; why did they have to flatten to begin with? So a grayscale image is 32x32 pixels, meaning it has 32 rows and 32 columns of pixels. By flattening them, you take all the pixels and put them in a single row, becoming 1024 pixel values. Not only that, but multi-demensional arrays take more memory while 1-D arrays take less memory.**

**I do have to express that I thought creating an algorithm to train a model would be more incredibly complex, although it's probably true that this example is just scratching the surface of what a real algorithm is composed of. I just thought it was interesting that this small piece of code can do impressive things.**

**I learned that data is extremely important to successfully train a model; the bigger the data, the better. This is because of all the variations of objects that are available; in parallel, advanced hardware is consequently needed.**

**When I tried to first type the code, it returned a long sentence of unreadable numbers, which consequently broke the IDLE environment. I learned that the labels (y\_train, y\_test) were in 2D and needed to be flatten with.ravel() to turn them into 1D arrays before filtering them.**

**Below are some of the notes that I took that mostly made me understand this project.**

With any newly created complex code, is it important to have libraries pre-installed and imported so that most of the work is done for us? In this case, we need to install Numpy, Matplotlib, TensorFlow, and Scikit Learn. We can do it simply by using:

**pip install numpy matplotlib tensorflow scikit-learn**

Now that they downloaded into our system, they need to be imported. As you will see, at the end of the import statement, you need to use an **alias** for the library imported to make the code cleaner and more readable:

import numpy as np

import matplotlib.pyplot as plt

from tensorflow.keras.datasets import cifar10

from sklearn.model\_selection import train\_test\_split

from sklearn.svm import SVC

from sklearn.metrics import classification\_report, accuracy\_score

**A screenshot of a computer program

Description automatically generated**

this is all CIFAR-10 SET UP

lots of code that I barely understand

**A screenshot of a computer program

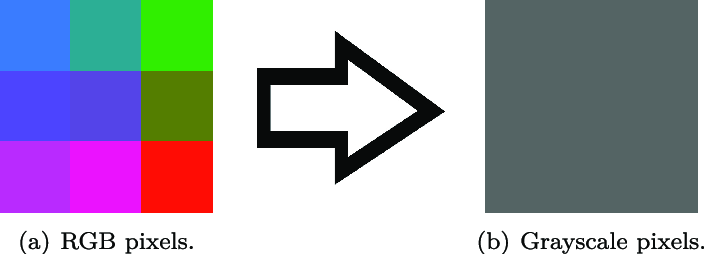
Description automatically generated**

COLOR TO GRAYYYY

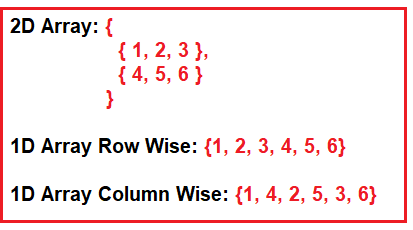
**CIFAR-10 is** vast amount of images that are used to train machine learning, it can be called a dataset because is a collection of data.



This data is then turned into **grayscale**. If the image we are processing has primary colors: red, blue, and green, they are converted to grayscale to avoid complexity, and the algorithm simply compresses the image to its bare minimum pixels.



Lastly, in the process of **flattening** the image, we are converting a two-dimensional image into a one-dimensional array so that it can be easily understood and processed by the neural network. When all of these steps are completed, we are ready to train the machine learning model.



Essentially, this picture shows flattening; it is converting a multi-dimensional tensor into a one-dimensional tensor. The order of this reshaping process is in row-major order; data is stored and accessed row by row. Single arrays are rows of pixels put sequentially into a long single row of pixels. (eitca.org)

Support Vector Machine (SVM) is a supervised machine learning algorithm.

SVM is a class in the Scikit-learn library. SVM looks for the best way to draw a line that divides one group from another. It does this by focusing on the most important points (the support vectors) and ensuring the line is as far away from both groups as possible.

It has many advantages, such as its effectiveness when the number of features is large, its memory efficiency because it uses a subset of support vectors in the decision function, and its flexibility with its kernel specification.

This part is where the gray images are classified into categorized in this case dog, cat, or ship.

A computer screen shot of a computer code

Description automatically generated

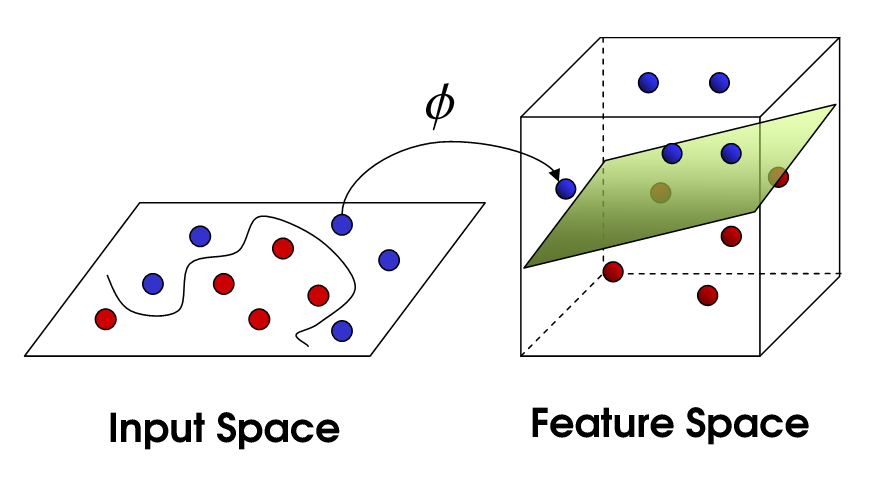
A screenshot of a computer program

Description automatically generated

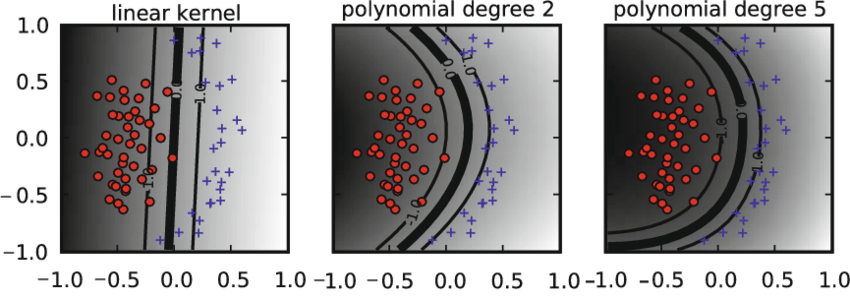
* **Hyperplane:** A decision boundary that separates different classes in the feature space. In 2D, it's a line; in 3D, it's a plane.
* **Support Vectors:** Data points that are closest to the hyperplane and influence its position and orientation. These points help in maximizing the margin of the classifier.
* **Margin:** The distance between the hyperplane and the closest data points from either class. SVM aims to maximize this margin.

There is 3 types of kernels:

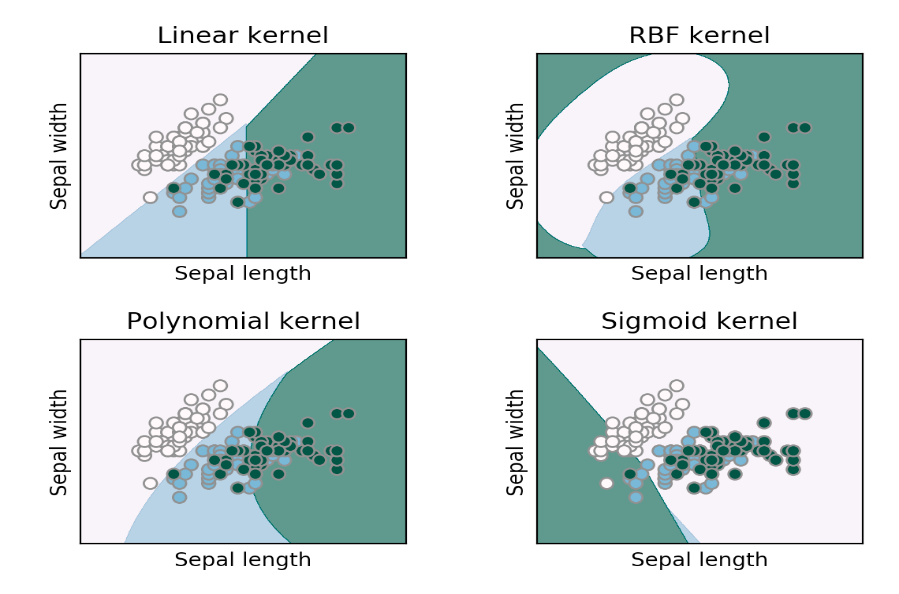
**linear kernal:** Meaning a straight line or hyperplane can separate the data. We can use this by using ***SVC(kernel = 'linear')***



**Polynomial kernel**: This data is not linearly separable, but a polynomial function of the input features can separate the data.



**Radial Basis Function (RBF) Kernel:** The data is not linearly separable, but mapping the data into a higher-dimensional space using a Gaussian (RBF) function can separate the data.



**WORKS CITED**

**“CIFAR-10.” *Wikipedia*, 29 Mar. 2021, en.wikipedia.org/wiki/CIFAR-10.**

**Academy, EITCA. “Why Do We Need to Flatten Images before Passing Them through the**

**Network?” *EITCA Academy*, 13 Aug. 2023, eitca.org/artificial-intelligence/eitc-ai-dlpp-**

**deep- learning-with-python-and-pytorch/neural-network/building-neural-**

**network/examination-**

**review-building-neural-network/why-do-we-need-to-flatten-images-before-passing-**

**them- through-the-network/#:~:text=The%20primary%20reason%20for%20flattening.**

**“Why to Use Grayscale Conversion during Image Processing?” *Www.isahit.com*,**

[**www.isahit.com/blog/why-to-use-grayscale-conversion-during-image**](http://www.isahit.com/blog/why-to-use-grayscale-conversion-during-image)

**processing#:~:text=Why%20is%20grayscale%20needed%20for.**

**Tabsharani, Fred . “What Is a Support Vector Machine? | Definition from WhatIs.” *TechTarget***

***WhatIs.com*, Loxz Digital Group, www.techtarget.com/whatis/definition/support-**

**vectormachineSVM#:~:text=A%20support%20vector%20machine%20(SVM)%20is%20**

**a%20type%20of%20supervised.**