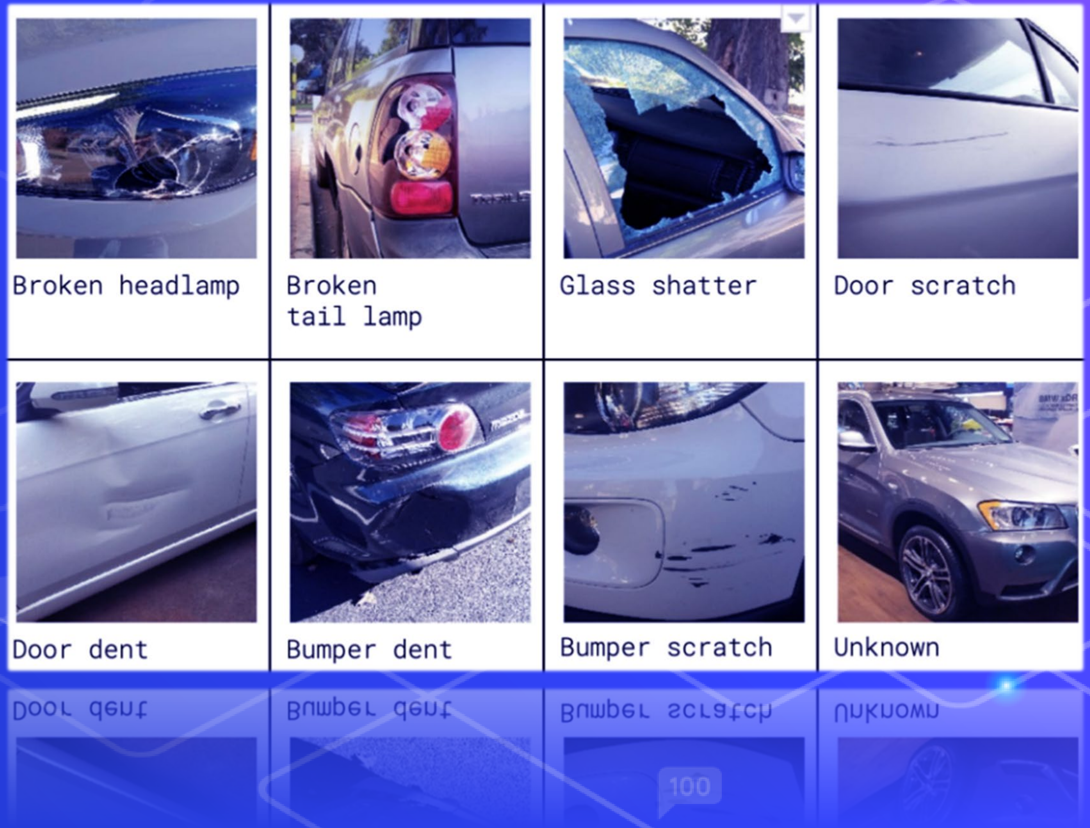
# Car Damage Assessment

Using Machine Learning

## The essence of the database:

The illustration show sample images from the various classes in the dataset.

Note that the unknown class contains images of cars that are in either pristine or wrecked condition.



**Door dent: 13% | Unknown: 28% | Head Lamp: 9% | Tail Lamp: 9% | Glass Shatter: 12% | Bumper Dent: 8% | Door Scratch: 13% | Bumper Scratch: 8%**

## Questions about the database:

- What are the success rates in **image** classification?
- Is machine learning effective in classifying a large number of classes?
- How much does modeling the image into a vector affect learning? (Flatten VS VGG)
- How much do image features affect learning?

## Algorithms

**CNN**
A concept of a neural network, Its main attributes may be that it consists of convolution layers, pooling layers , activation layers etc.

CNN

**SVM**
Works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong

SVM

KNN

works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression)

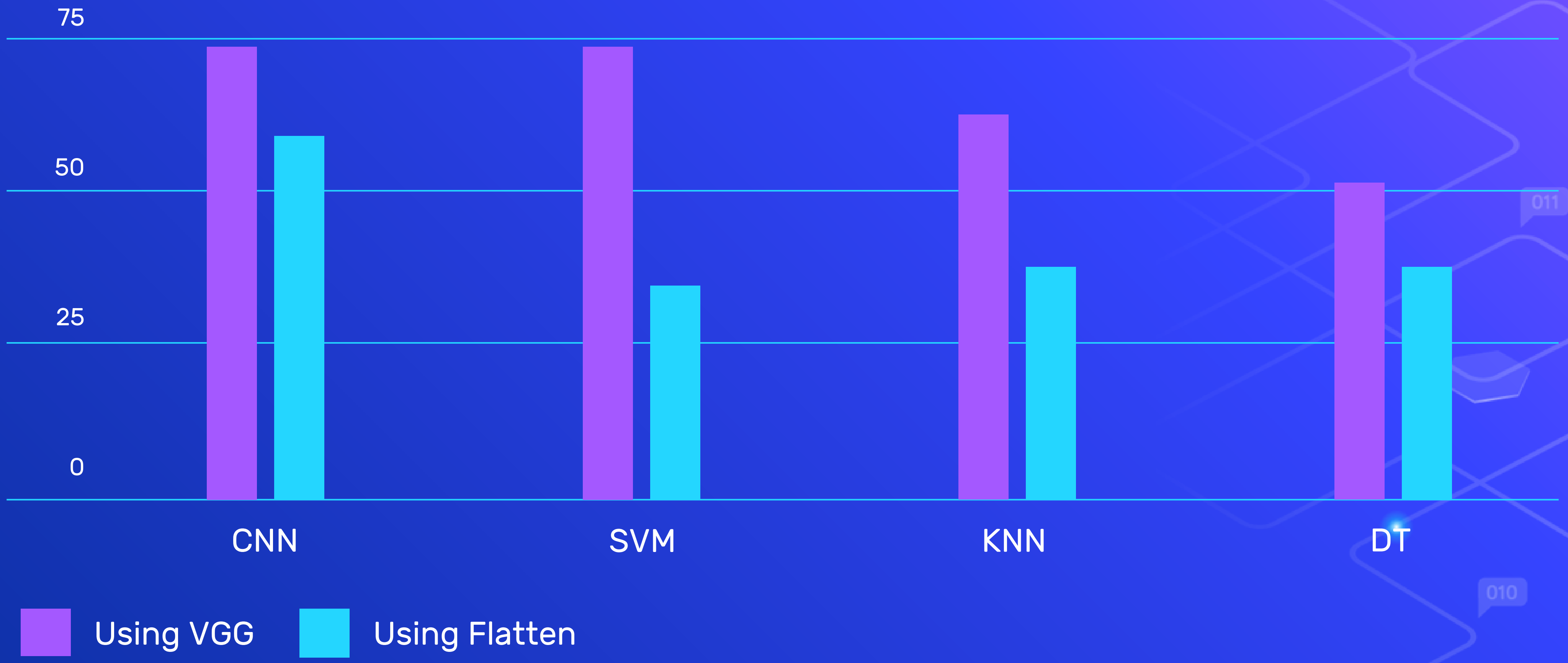
**KNN**

DT

DT use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes

**DT**

## Results



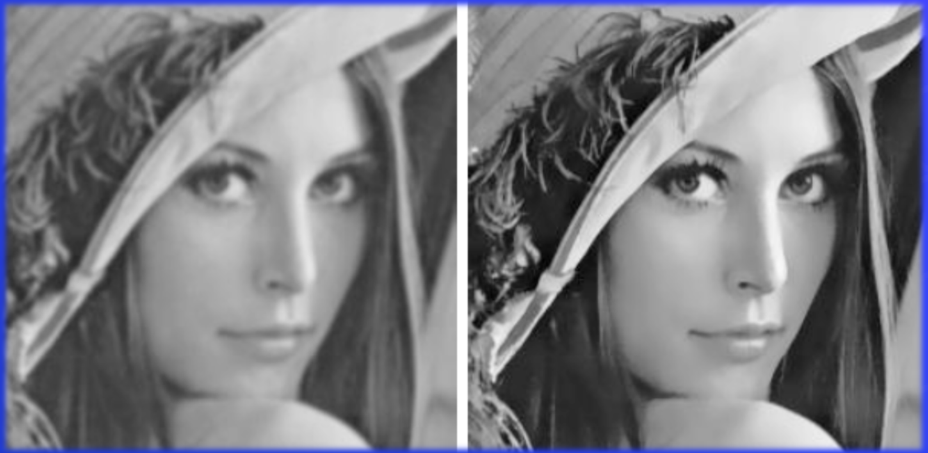
## Main Challenges

- **Loss of information obtained from the image columns as a result of performing Flatten (converting the image to vector)**
  **<u>Overcoming</u>:** By using VGG
- **Classification into a larger number of classes**
  **<u>Overcoming</u>:** By using larger images that provide more information (224X224)

## Techniques

- **Bilateral filter**

  Noise removal while maintaining edges

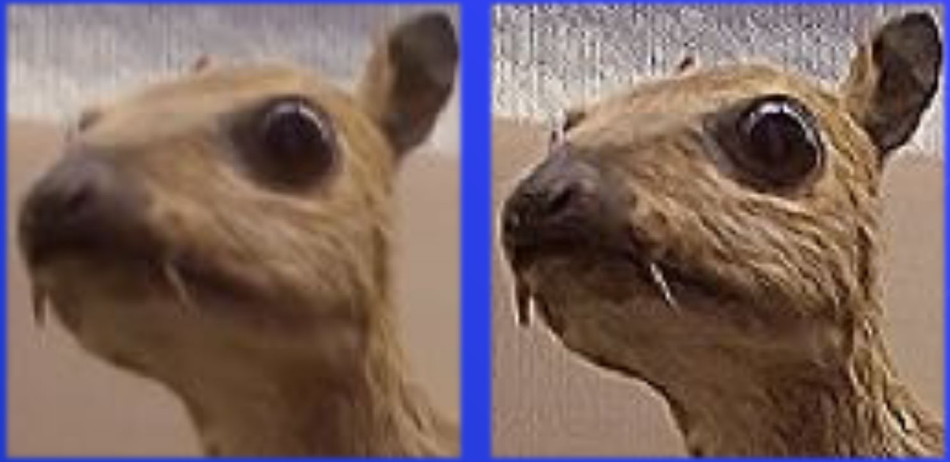


**Gaussian Filter** **Bilateral Filter**

## Techniques

- **Sharp Via Laplacian Of Gaussian**

  Sharpening edges by subtracting the second derivative of the image from the original image



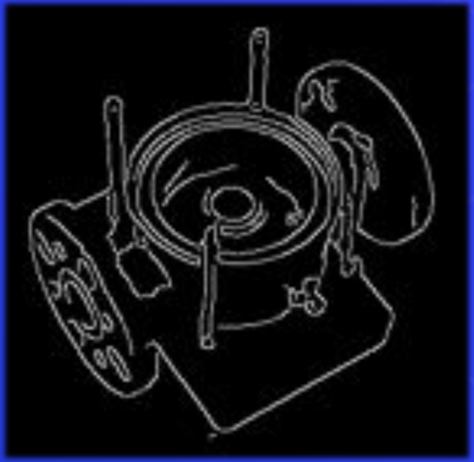
**Original** **Sharpen**

## Techniques

- **Canny**

  Finding the edges by the first derivative of the image and filtering the results with the assumption that the edges are long and connected together (using angles)

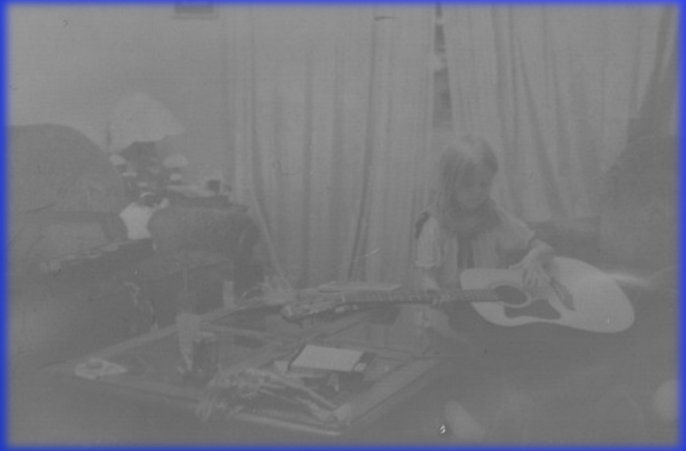
**Original**

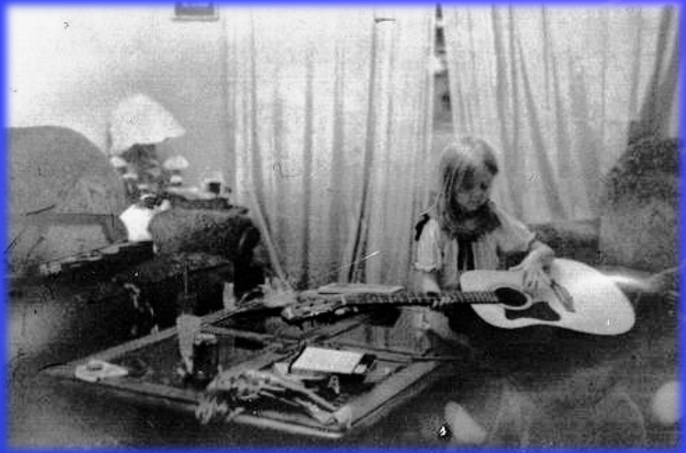

**Edges**

## Techniques

- **Histogram Equalization**

  Increasing image contrast (larger changes between pixels)


**Before**

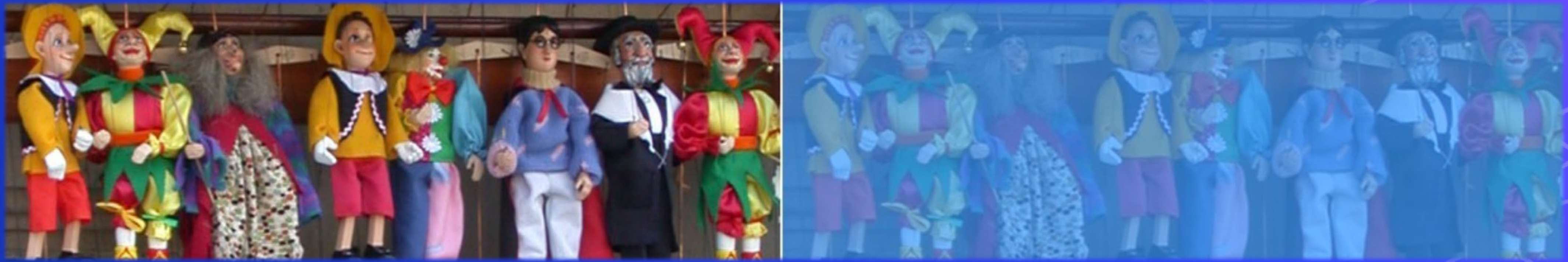

**After**

## Techniques

- **White Balance**

  Corrects the color appearance of the image

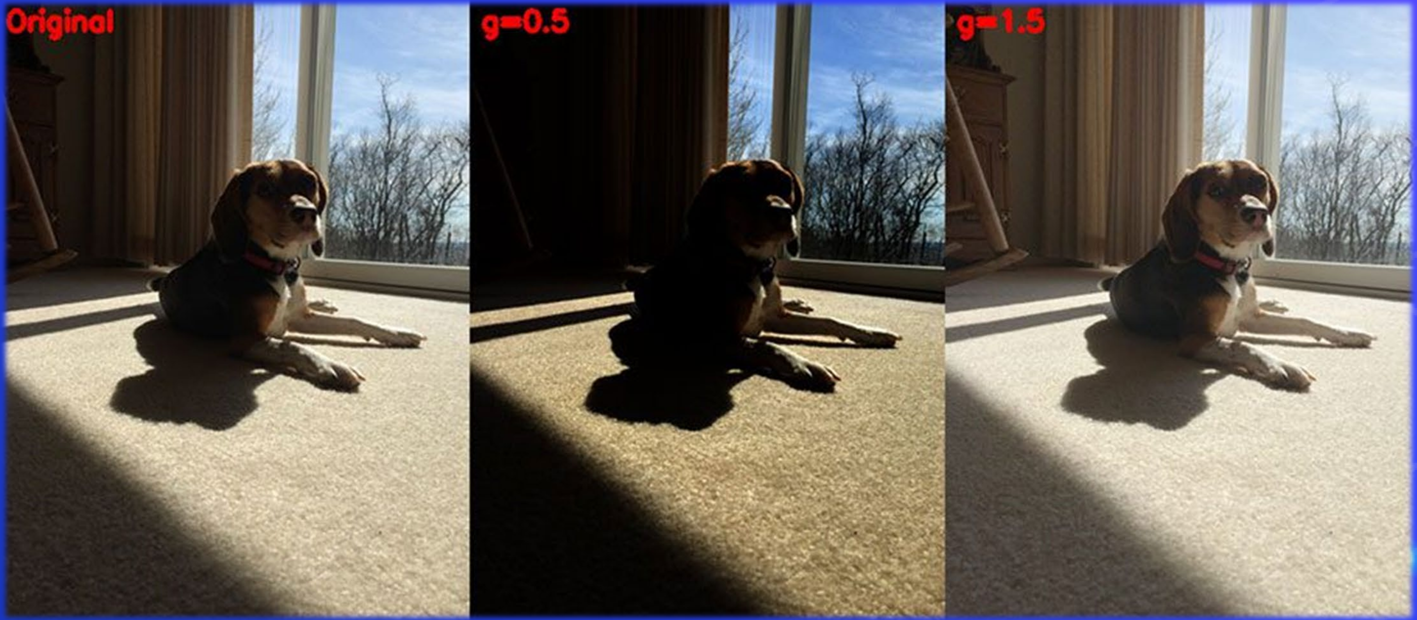
  based on the white color



## Techniques

- **Gamma Correction**

  Change the brightness of the image

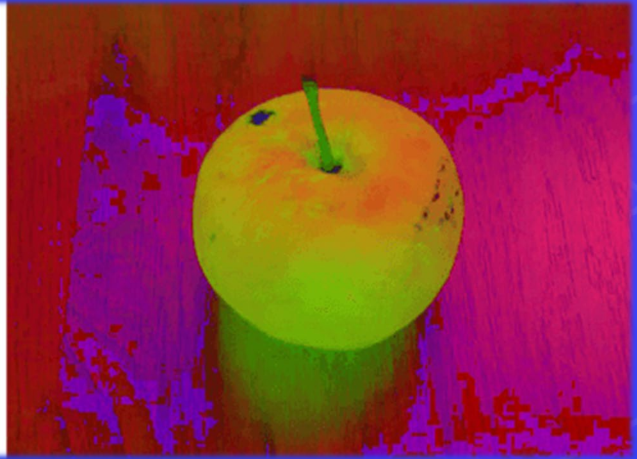


## Techniques

- **HSV**

  Color space that defined by hue (color range), saturation (darkness range), value (brightness range)

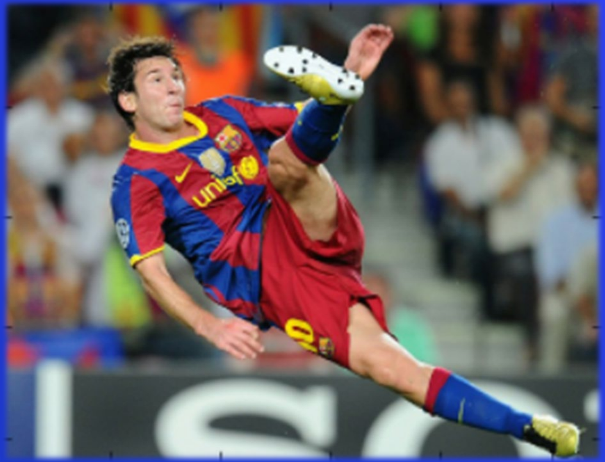
**RGB**

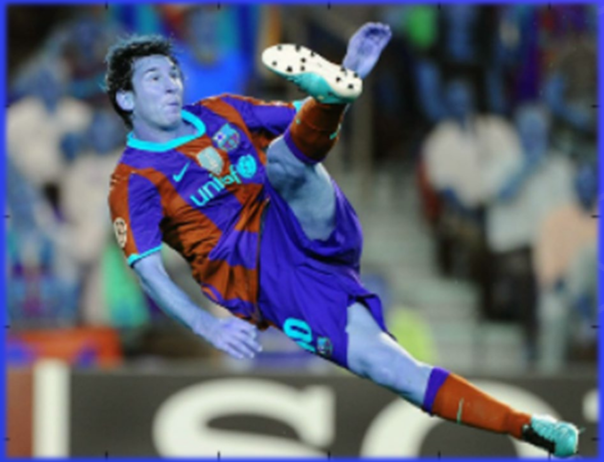

**HSV**

## Techniques

- **BGR**

  Color space similar to RGB color space but in reverse layout

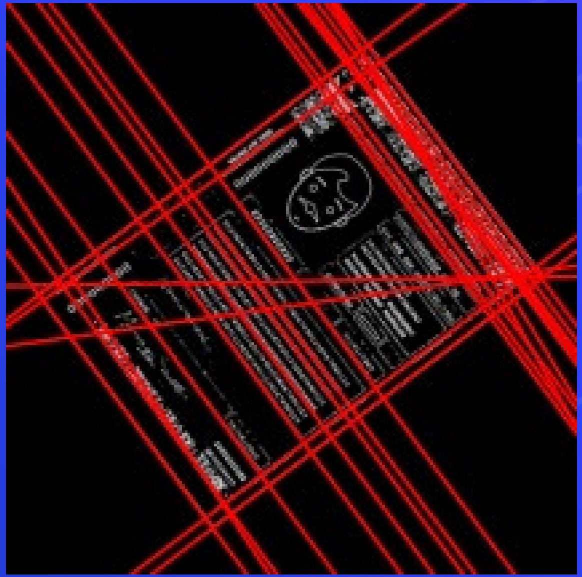

**RGB**


**BGR**

## Techniques

- **Hough Line Transform**

  Finding a parametric shape (characterized by parameters like line) by mapping each edge pixel to a new space whose dimensions represent the shape parameters

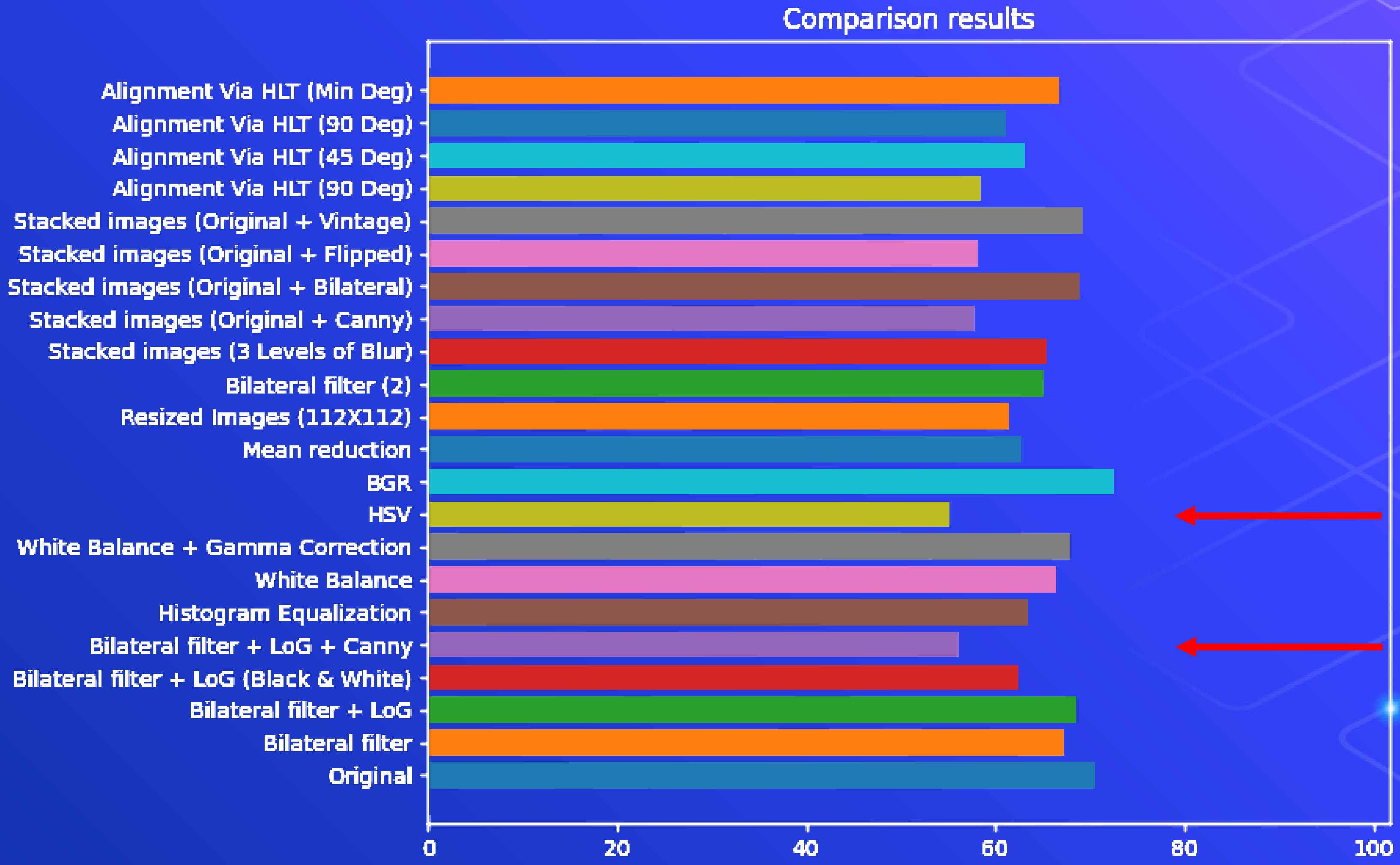


## Techniques

… And more tweaks like:

- Mean reduction
- Resized Images (112X112)

## Techniques Results



## Analysis Of The Techniques

- **Hough Line Transform  -** Loss of information of the image corners as a result of rotation
- **Using Canny –** Over-adjustment to a specific image as a result of highlighting the objects
- **HSV -** Loss of color distribution (to a smaller number)
- **Histogram Equalization** - Loss of color matching between images due to minimum and maximum value
- **Black & White** - Loss of information due to image colors
- **Image Reduction** - information loss / aliasing

## Analysis Of The Algorithms

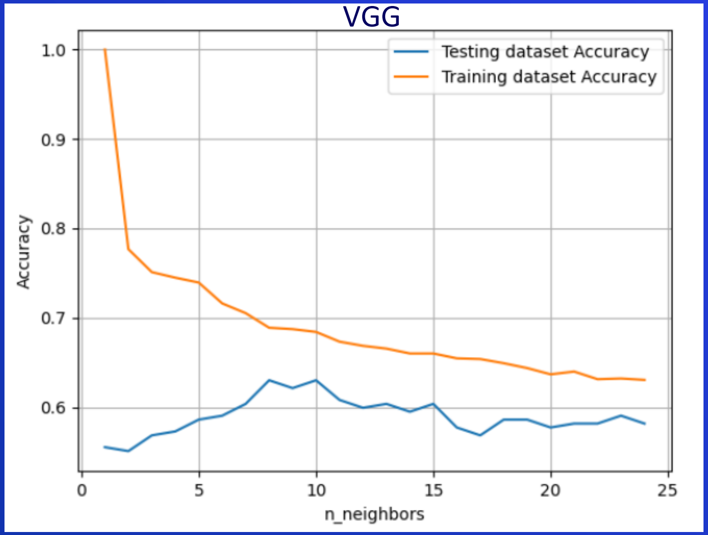
- Rising trend in success rates
- How each algorithm behaved?
- Compare the different algorithms

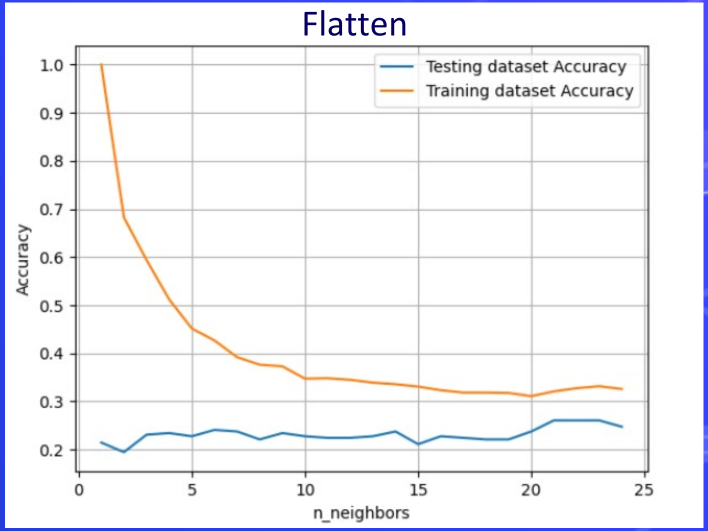
## Rising Trend In Success Rates

- By simply randomly guessing, you should be able to reach ~ **12.5% accuracy** (since there are eight class labels).
- After running the algorithms (SVM, DT, KNN) with Flatten we get ~ **35% accuracy** (+22.5%)
- After running CNN algorithm with Flatten we get ~ **55% accuracy** (+42.5%)
- After running the algorithms (DT, KNN) with VGG we get ~ **65% accuracy** (+52.5%)
- After running CNN & SVM algorithms with VGG we get ~ **75% accuracy (+62.5%)!!**

## K-Nearest Neighbors Algorithm

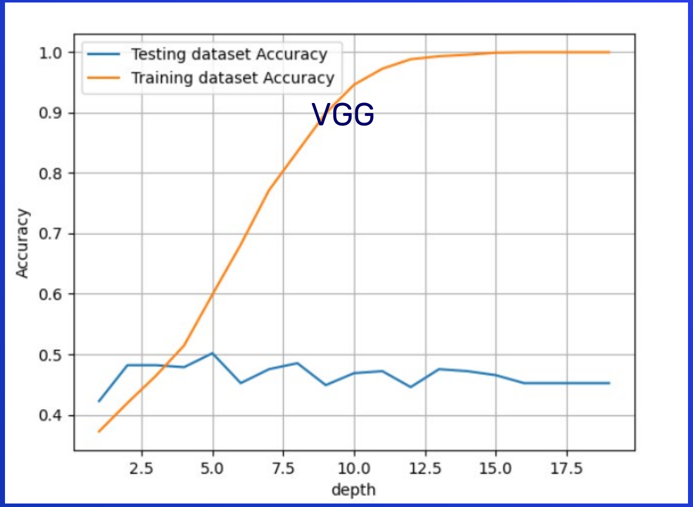
- k should be wisely selected. (k~8-10 is the best result with VGG)
- Success rates are very low with Flatten because we're losing information from the image columns.

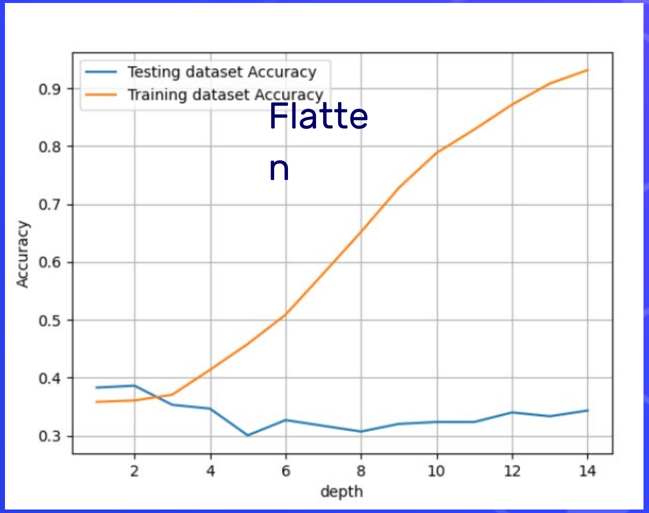




# Decision Tree Algorithm

- Tree may grow to be very complex while training complicated datasets.
- VGG reaches high percentages in larger trees compared to Flatten





## k-NN vs DT

- In both with Flatten we can demonstrate there is an underlying pattern to the images for both raw pixel intensities.
- Decision tree is faster due to KNN’s expensive real time execution.
- KNN gives better results than DT after VGG because processing is done for numerical data

## SVM vs DT & k-NN

- SVM gives better results than DT & KNN after VGG
- Decision trees are better for categorical data and it deals collinearity better than SVM.
- SVM take cares of outliers better than KNN.
- SVM outperforms KNN when there are large features and lesser training data.

## SVM vs CNN

- SVM can perform better than NN when there are limited training data and many features. NN needs large training data for sufficient accuracy.
- Multi class classification requires multiple models for SVM, whereas NN can do it with a single model.

## Credits

Authors:

- Almog Jakov
- Itay Rafee

Visit Us:

https://github.com/AlmogJakov

https://github.com/itay-rafee

Made Using PowerPoint®

# Thanks!