Effect of converting 3d to 2d image and applying KAZE feature descriptor on CNN

PROJECT BY:

ABHIJIT TALLURI (11527014)

KAVYA SRI ACHANTA (11591814)

VENKATA NAGA PRASANNA KUMAR BITRA (11525657)

PRASANTH DAMARLA (11525662

Goals and Objectives:

1) Motivation:

   The motivation of the project is to understand the effects of converting a 3D image to 2D image and also understanding the KAZE detector feature on the CNN (Convolutional neural network) model.

2) Significance:

   The significance of the project using the RESNET framework. ResNet-50 is a CNN that is 50 layers deep. The RESNET framework always stood on the top. So, we are using the RESNET model with KAZE filter, which makes the model more reliable.

3) Objectives:

   The objective of the project is to extract the features by KAZE feature. And converting the 3D image to 2D without using the gray scale. For converting 3D to 2D we are using matrix summation of RGB.

4) Features:

   The feature here we are using is KAZE. KAZE feature is a 2D feature detection and the description method that works completely on the nonlinear scale space.

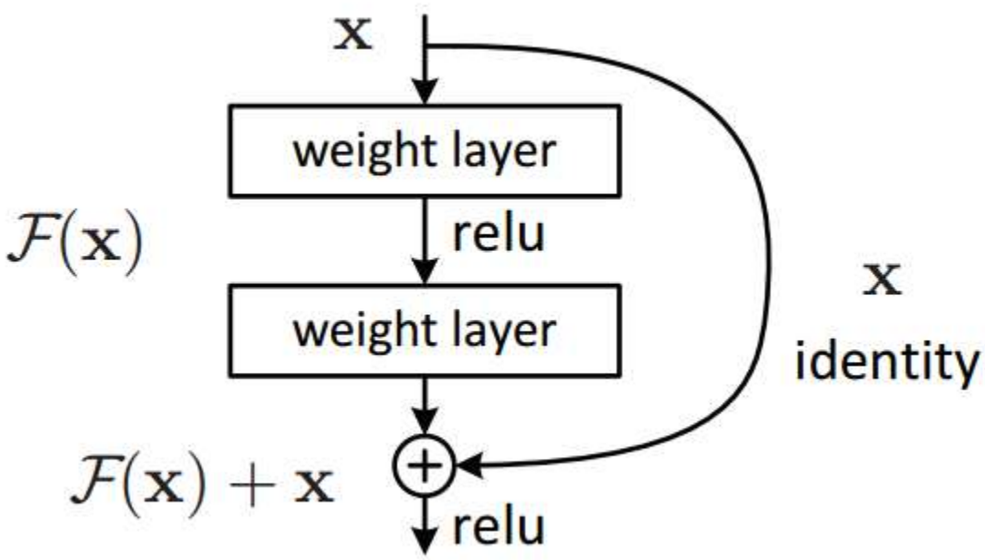
Introduction:

This project is mainly concentrate on the effects of converting 3D to 2D image and applying the KAZE feature descriptor on the CNN model. KAZE feature is a 2D feature detection and the description method that works completely on the nonlinear scale space. The 3D image is converted into 2D image by matrix summation without using the gray scale. Which will help us to get the better results. And the model is trained using the prediction model.

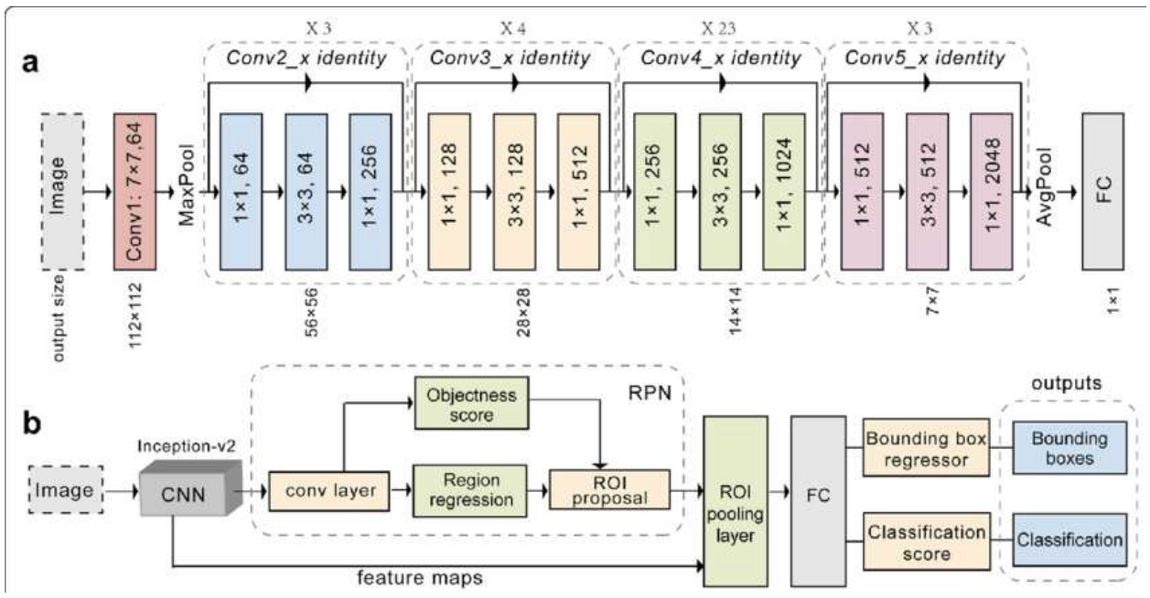
Related Work (Background):

This project is developed on the Residual Network. It will give you the better speed and the accuracy. The Residual network is designed for the deterioration problem. Which makes frames much easier to train the network than the previous frameworks. Here 152 layers are used in the framework. This layers are similar to the CNN (Convolutional Neural Networks). They are few level to reduce the framework depth.

Architecture and workflow:



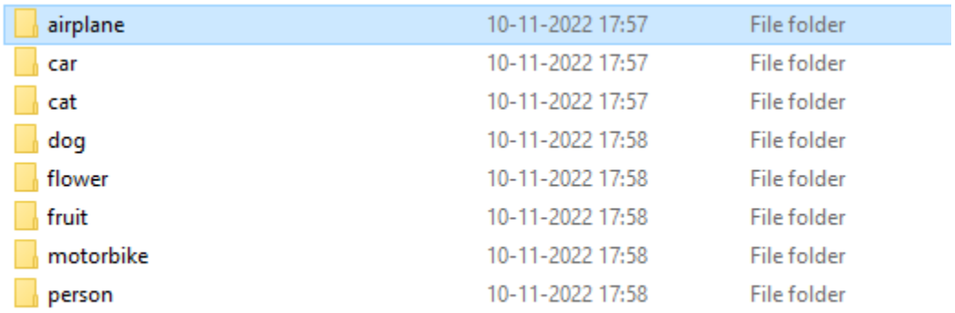
RESNET is used to improve the accuracy of the model and to overcome the drawbacks in the CNN (convolution network). In RESNET it takes the image as the input and sends the feedback to the processed image. The f(x) is the output of the CNN for the original image x the output of the RESNET is h(x) which is f(x) + x.

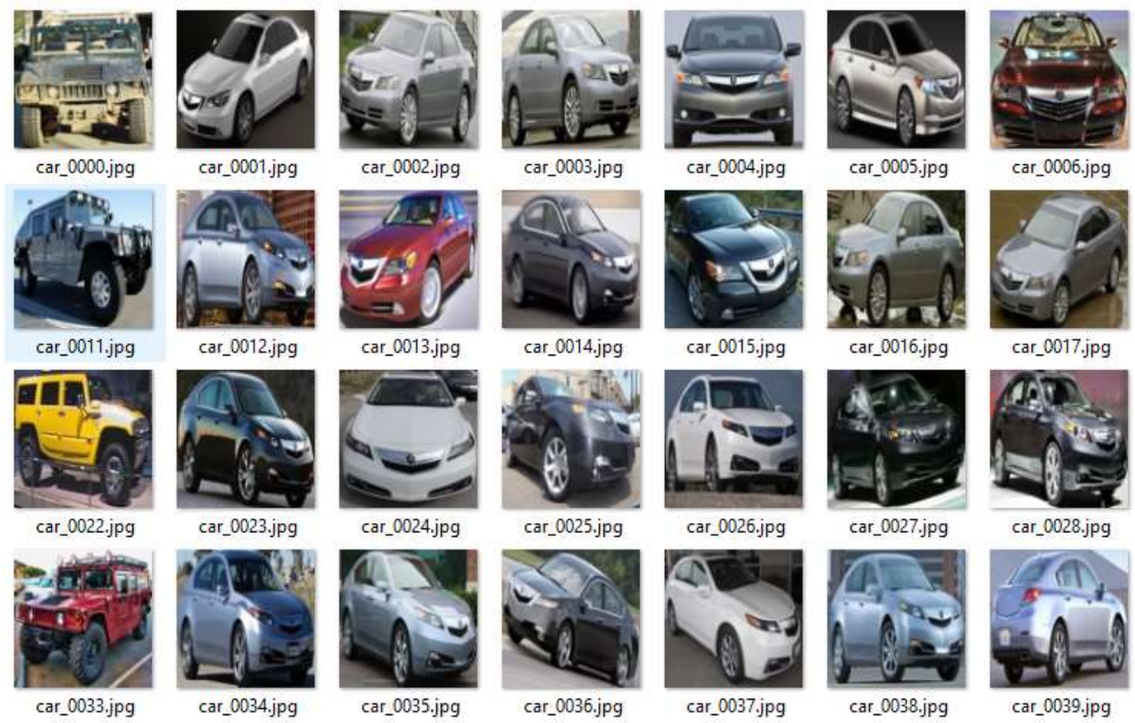


The above the work flow diagram of RESNET.

Data set:-

In the project we are using the dataset from Kaggle. The dataset contains the airplane, car, cat, dog, flower, fruits, motorbike and person images. Using this dataset, we can train the model to get the accuracy.





Dataset of the cars

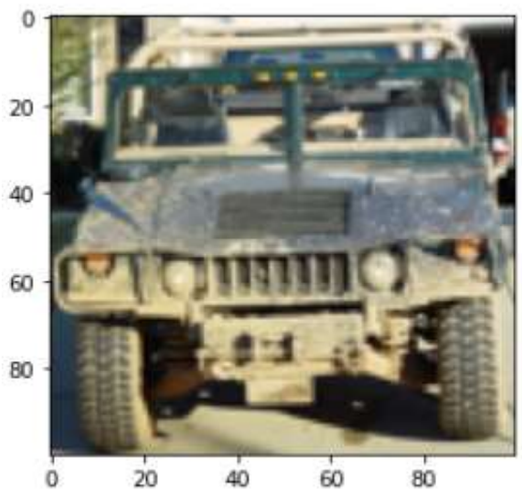
Dataset for the motorbikes

The dataset is divided into test and train datasets.

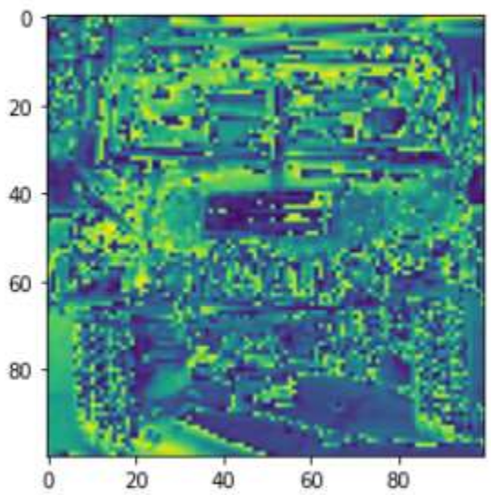
DETAIL DESIGN and FEATURES:

1. **KAZE**: KAZE: The technique identifies and characterizes 2D properties in a non-linear scale space to increase separation and placement accuracy. A non-linear scale space is constructed from an input image by discretizing its scale space in a series of octaves and sublevels while keeping the original image's resolution. Then, from pixels, translate the collection of discrete scale levels in time units I. The non-linear scale space is then built iteratively using additive operator splitting (AOS) and variable conductance diffusion techniques to reduce noise and retain the borders of the objects in the image.

   To find 2D features, the scaled normalization determinant of a Hessian matrix, produced at various scale levels with maxima as important aspects, is employed. By figuring out the keypoint orientation, a scale- and rotation-invariant descriptor is produced, and this serves as the description. The computational cost of this approach is high.



Original image



KAZE filter

Analysis:-

The goal of the project analysis is to identify characteristics using KAZE features. likewise converting a 3D picture into a 2D picture without using grayscale. We employ RGB matrix summing to translate 3D into 2D. The significance of utilizing the RESNET framework for this project. ResNet-50 is a 50-layer CNN. Always at the top was the RESNET framework. To boost the model's reliability, we therefore combine the RESNET model with the KAZE filter.

Implementation:-

Algorithm/Pseudo code and Explanation:

1) Reading the image and applying the filter to the image

```
def single_channel_image(img):
 (B, G, R) = cv2.split(img)
 return B+G+R

plt.imshow(single_channel_image(cv2.imread('/content/down_imagenet
/train/car/car_0000.jpg')))
```

2) Applying RESTNET model on the dataset.

```
resnet50 = keras.applications.resnet50
conv_model = resnet50.ResNet50(weights='imagenet', include_top=False)
conv_model.summary()
```

3) Printing the accuracy of the filter and adjusting the epochs accordingly

```
Full_model.fit_generator(train,validation,workers,epochs)
Full_model.predict()
```

4) Appling the KAZE filter on the whole dataset.

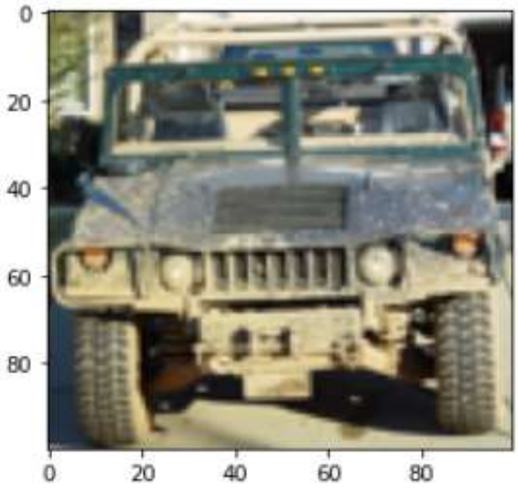
```
def preprocess_input(img): #appling on the dataset
img = np.array(img, dtype=np.uint8)
query_img = single_channel_image(img)
query_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
akaze=cv2.AKAZE_create()
queryKeypoints, queryDescriptors = akaze.detectAndCompute(query_img,None)
img = cv2.drawKeypoints(img,queryKeypoints,outImage = None,color=(255,0,0),
flags=0)
return resnet50.preprocess_input(img)
```

GitHub Link : - https://github.com/Kavyaachanta0808/fe

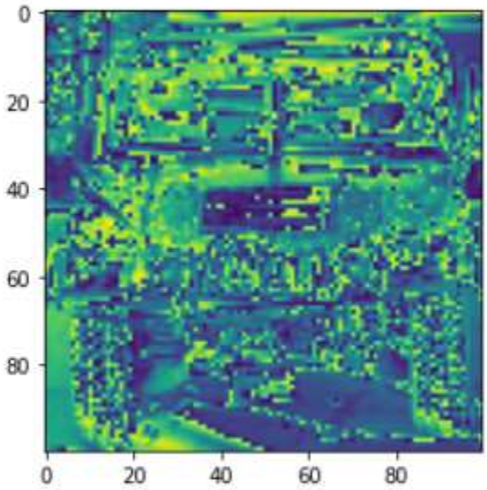
Preliminary Results:

On applying different filters to the dataset and feeding the data to the model provides the accuracy and loss for both training and validation phases.

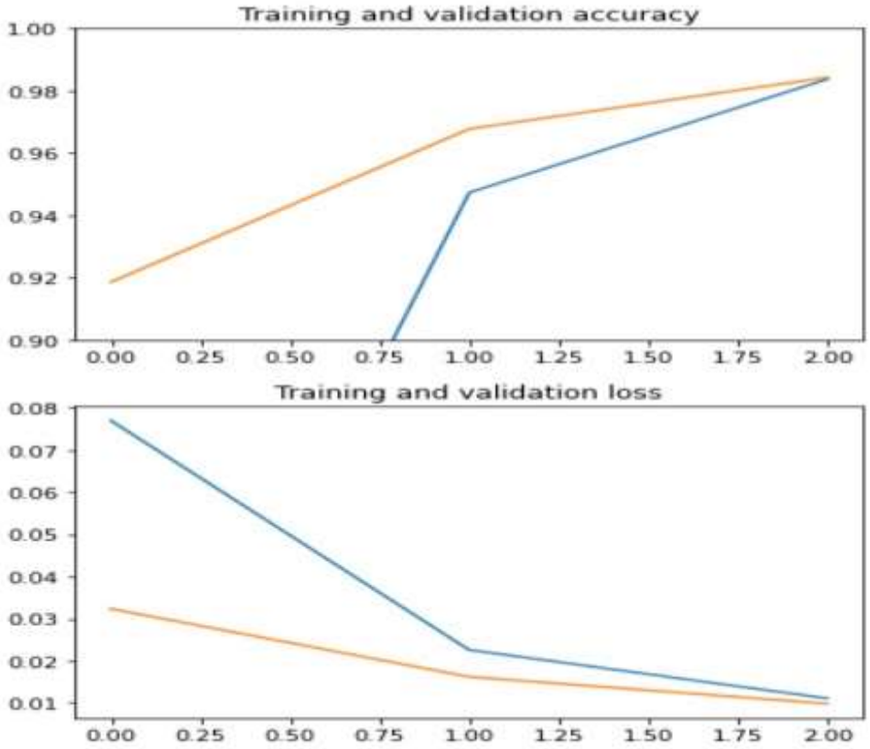
KAZE:-



Original image



KAZE filter



	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
1	1.00	1.00	1.00	1
10	1.00	1.00	1.00	1
11	1.00	1.00	1.00	1
12	1.00	1.00	1.00	1
13	1.00	1.00	1.00	1
14	1.00	1.00	1.00	1
15	1.00	1.00	1.00	1
16	1.00	1.00	1.00	1
17	1.00	1.00	1.00	1
18	1.00	1.00	1.00	1
19	1.00	1.00	1.00	1
2	1.00	1.00	1.00	1
20	1.00	1.00	1.00	1
21	1.00	1.00	1.00	1
22	0.00	0.00	0.00	0
23	1.00	1.00	1.00	1
24	1.00	1.00	1.00	1
3	1.00	1.00	1.00	1
4	0.00	0.00	0.00	0
5	1.00	0.50	0.67	2
6	1.00	1.00	1.00	1
7	1.00	1.00	1.00	1
8	1.00	1.00	1.00	1
9	0.00	0.00	0.00	1
accuracy			0.92	24
macro avg	0.88	0.86	0.87	24
weighted avg	0.96	0.92	0.93	24

```
[[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1]]
```

Project Management:

- **Implementation status report**

Work completed:

- Description - We have finished converting 3D vectors to 2D vectors, applying Kaze feature descriptors, and passing through CNN, but we still need to compare the throughput and efficiency of the OK base CNN network.

Work to be completed :

- Description - We are working on some of the feature selection methods to be reworked. Aside from that, we must work on model training in TPU because GPU resources are insufficient. Following training, we must focus on improving accuracy.