TRAFFIC SIGN CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

VENUBABU RACHAPUDI¹, M SITHA RAM², P DILEEP³, T.L. DEEPIKA ROY⁴
¹Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
²Department of CSE, Lakireddy Bali Reddy College of Engineering, Mylavaram, Vijayawada, Andhra Pradesh, India
³Department of Computer Science and Engineering, Malla Reddy College of Engineering and Technology, Kompally, Hyderabad, Telangana, India
⁴CMR Technical Campus, Hyderabad, Telangana, India

ABSTRACT:
Traffic signs are a very crucial part in ensuring the safety of people travelling on road and in avoiding accidents. Often, we find various types of signs on the roadsides like speed limit signs, stop signs, yield signs etc. As humans, we can recognize these signs by looking at them, but computer systems cannot interpret what they mean. Traffic sign classification project aims at classifying these traffic signs automatically into their respective types using Convolutional neural networks. This project can then be implemented in autonomous driving vehicles, helping them in making better decisions and providing safer drives. In this project, we use the German Traffic sign dataset which comprises of traffic sign images of forty-three classes. Along with convolutional neural networks, we also use various preprocessing techniques like data augmentation, grayscaling, normalization for enhancing the feature recognition aspect of images.

Keywords: Traffic signs, Classification, Convolutional neural networks, Preprocessing.

1. INTRODUCTION
Traffic sign classification is an artificial intelligence project to identify or divide the traffic signs into their respective classes [7][8]. Traffic sign classification involves collecting and analysis of traffic signs images in the early stages. The traffic signs are images captured on the road by static people or moving automated cars and vehicles. This classification system is extremely useful in automated vehicles that use sensors and cameras to analyze their surrounding environment and take decisions. It is very important to classify the seen traffic signs along the road into their specific classes like “60 kmph speed limit” sign or a “school zone” sign.

The dataset used is the benchmarked “German Traffic Sign Recognition” dataset. It has traffic sign images for training and testing purposes. A very crucial part of the project is pre-processing the dataset to maximize accuracy[9]. We use techniques like normalization and data augmentation to make the image more contrasted and clearer and to make the system more robust. Then we use Convolutional Neural networks to build the model. CNN is used in the system because it is the best fit for analyzing image datasets[10][11]. We use VGG architecture for the model. Images are arrays of pixel values. Pixel values range from 0(black) to 255(white) To perform analysis and training, it is essential for all images to be uniform in size. All images are 32(width) x32 (width) pixels in size.
A channel in this context is the grayscale image of the same size as a color image, made of just one of the primary colors like red etc. It is very important to have images containing three color channels to have better analysis chances[12]. All images have 3 color channels: Red, Green and Blue as shown in the Fig1. There are 43 classes in total of the traffic sign images (e.g., Speed limit 20kmph, No entry, bumpy road etc.).

II. LITERATURE SURVEY

P. Sermanet and Y. LeCun proposed a multi scale convolutional network for traffic sign identification[1]. Advantage is its multiscale feature learning is used to run several CNN models with varying contextual input size. Limitation identified is that the CNN did not consider the position and orientation of the object.

Stallkamp et.al. discussed Recognition of German Traffic Sign[2]. They provided Multiple stages of feature extraction provide hierarchical and robust representation. Model is suffered with Spatial variance in the input data which affect the model performance. The scope we identified is More diverse training set deformations can also be investigating such as brightness etc.

Miura et.al proposed on-line traffic sign recognition which uses Recognition algorithmsby intensively using built in functions of an off the shelf image processing board[3]. They come up with Easy implementation and fast recognition. But identified that Limited feature extraction and spatially invariant.

A. Wong et.al. Micronet which achieves a good balance between accuracy and model size as well as inference speed[4] and which takes lot parameters into consideration which leads to computationally complex.

Alexander Shustanov et.al proposed CNN for real time traffic signs with high recognition rate and fast execution[5]. the convolutional neural networks had enhanced most of computer vision tasks.

Wenhui Li et.al. given a new CNN model. Feature extraction, compared with the traditional CNN method, has higher accuracy, smaller models and easier training[6]. We observed that we can add residuals, which can effectively avoid gradient disappearance, improve accuracy, and light weight models.

III. THEORETICAL ANALYSIS

Preprocessing is an important part of developing a robust classification system. As our dataset contains images and we are using convolutional neural networks the preprocessing required is less but significant. Convolutional neural networks work on finding the important features of an input image[13]. When preprocessing is done on the input image, its sharpness and contrast is generally increased allowing for better feature recognition. The preprocessing techniques we have used are shuffling, grayscaling, normalization[14].

In general, we shuffle the training data to increase randomness and variety in training dataset and the model becomes more stable. We use stratified random sampling in the project. In stratified random sampling we use the technique of dividing the entire information present into small sub parts called strata[15]. To make this process efficient, the shuffling is done based on the attributes of the data elements. To obtain a sample dataset that is equivalent to other sub parts of the dataset based on characteristics, use stratified random sampling. It includes...
the process of diving the full information set into homogenous groups called strata. These strata can be used as training, validation and testing sets. Stratified random sampling is more effective than simple random sampling, where the the populating of subgroups is done randomly, so data might get skewed.

Gray scaling is the process of converting an image to grayscale from its previous color schemes like RGB, HSV, etc. It includes color shades between complete black and complete white. When there are multiple color channels, there will be multiple dimensions to the image but once the image is converted to greyscale, it will have only one dimension[16]. Dimensionality reduction will automatically lead to decrease in complexity of the algorithm. For e.g. When we train neural network on images of size 10x5x3, the input layer will have 150 nodes but in grayscale it will only have 50 nodes. A lot of the algorithms are built to work specifically on grayscale images like canny edge detection algorithm. It is pre-implemented in OpenCV library and functions only on grayscale images.

IV. PREPROCESSING TECHNIQUES

Image normalization is a typical process in image processing that changes the range of pixel intensity values. When an input image is supplied to the algorithms, its generic job is to convert it into a set of pixel values that make more sense and are easy to interpret. In this work, we will perform a function that produces a normalization of an input image (grayscale). Once it is converted, we try to understand these values which are between 0 and 255[17]. Hence we can interpret the dark or blurry images more efficiently.

The normalization of an input image is performed according to the formula

$$\text{Output channel} = 255 \times \frac{\text{Input channel} - \text{min}}{\text{max}-\text{min}}$$

If we are using a grayscale image, we need to normalize using only one channel. We then subtract each image by the mean of the dataset and divide by its standard deviation to centralize the distribution. This process is performed to make the model treat all images equally. The resulting images look as follows:

![Fig.2 Example of normalization on the images](image)

Local histogram equalization is used to increase the contrast in input images. It does so by spreading out the most repetitive intensity values in the given image. When we have a lot of real time images with low contrast in our dataset, this method comes in handy. The library used here is “skimage”. It provides for the application of local histogram equalization on training images.

A good classification algorithm should be able to classify images even when they are in different orientations. Many times, in real world datasets, images are distorted or found in irregular angles. This dataset, GTSRB also has some imbalance across the forty-three classes. All these issues can be solved using data augmentation because it takes the following actions[18].
We have augmented each of the classes present by ten percent. There still might be some imbalance induced bias in our model but we have concentrated on augmentation as we do not want to increase the training time by making the dataset bigger and bigger.

V. MODEL ARCHITECTURE

For building the model, we use Convolutional Neural Networks. A Convolutional Neural Network also called briefly as “ConvNet” or “CNN” is one of the deep learning algorithms which can take in an input image, assign importance like variable weights and biases to different aspects in the image and be able to differentiate one image from another.

Dataset Description:
The dataset is “German Traffic Sign benchmark dataset. Benchmarking is the technique of measuring how good or effective a process or service is. The process involves finding a good or the best comparable process or service in the same domain and setting it up as a standard. The process at hand is then compared to the standard and evaluated.

The dataset is split into training, validation and test sets. A training dataset is a set of data used during the learning cycle and is used to fit the parameters like weights of the model for example, a classifier. The bigger the dataset, more is the chance for better accuracy and precision. The dataset has 34,799 images in the training set. The validation dataset is the set of data used to give an accurate evaluation of a model performance on the training dataset while fine tuning model hyperparameters. Validation dataset is used to frequently check the model fit. This dataset provides 4,410 images in the validation set.

Proposed Methodology:
The architecture of a ConvNet is similar to that of the connectivity pattern of neurons in the human brain and was inspired by the organization of the visual cortex. In the receptive field, neurons react to only those stimuli in the restricted space. The visual area is the collection of such individual spaces/areas.

The convolutional layer provides us the flexibility to build upon previous layer’s outputs. It is mainly used to identify the important features of input images. The features can be high level and low level. Low level features include edges or colors and these are generally captured by initial convolutional layers. The following convolutional layers are responsible for identifying high level features. By using these layers, we get a holistic understanding of the image.

The pooling layer’s primary job is to identify the important or dominant features that the model might dismiss due to variations in position or arrangement of the input image. In doing this, the pooling algorithm also reduces the dimension of the extracted feature in turn leading to the requirement of lesser computational power.

A fully connected layer is one in which every neuron is connected to every other neuron. It is a good method to identify the nonlinear connections in the high-level features. These high-level features are the outputs given by preceding convolutional layers.

At each level of the training process, dropout algorithm sets the neurons of some hidden layers to zero thus making their representations or feedbacks ineffective. This is done to avoid overfitting in the model. Generally overfitting happens when the training dataset is small or not diverse enough.

VI. EXPERIMENTAL RESULTS

In the building of the model, we have started with a raw unprocessed dataset. The model initially had convolutional, pooling and fully connected layers.
When the model is trained on raw dataset, we have observed the accuracy to be 83 percent. It is evident that by processing the data accuracy can be increased. So, we have tried training the model on grayscaled data and achieved an accuracy of 92 percent. Then we have just normalized the dataset and achieved an accuracy of 93 percent. So, by combining both the methods of grayscaling and normalization, an accuracy of 96 percent has been achieved. Fig3 and Fig4 shows probabilities on custom images.

Further by adding histogram equalization, data augmentation techniques and dropout layers to the model, we have successfully achieved 97.5 percent accuracy.

**VII. CONCLUSION**

From the experimental results mentioned above, we find that when we use raw images without any preprocessing, the accuracy is relatively less. This is because not all images are clear in the dataset. Some of them are too dark due to underexposure, some of them are too light due to over exposure to light. There are also blurry images and images where the traffic sign is not properly erected for e.g., being tilted to the side. These irregularities have prevented our model from predicting accurately. Once we have applied grayscaling on the dataset images, the accuracy has increased because now the complexity of images has decreased. Earlier the images had three color channels, so there is an extra dimension to be considered but now, the image is just a single vector with different intensity values. This has enabled our model to work better. Further we applied the techniques of normalization and the accuracy has increased by a small value. This is because normalization will change the scale of intensity values and is helpful in cases where the images have low contrast and are indistinguishable. Histogram equalization also increases the contrast in images. So, by combining these algorithms we were able to increase
the accuracy by three percent. The accuracy has risen to ninety seven percent as we have also added dropout layers to the model architecture. The dropout layers reduce any overfitting of the model on the training dataset by strategically nullifying the effect of neurons at certain levels, this will allow room for error and thus model will learn much better from these kinds of predictions.

REFERENCES:


