Math/Comp – 546

Pattern Recognition

Project Report

Title: Automatic Number Plate Recognition (ANPR)

Authors: Rishikesh Patil, Marrshal Ben Manuel

Introduction

ANPR systems have evolved as significant tools in the realm of computer vision and pattern recognition. These systems are intended to identify, locate, and recognize license plates from pictures or video feeds automatically. Law enforcement, traffic control, toll collecting, and parking management have all benefited from ANPR technology.

The major goal of this project is to create an efficient and reliable ANPR system capable of consistently identifying and analyzing license plates from various sources. The system seeks to deliver real-time and automated license plate identification capabilities by employing sophisticated algorithms, machine learning approaches, and image processing technologies.

The importance of the ANPR project stems from its potential to transform a variety of fields. In law enforcement, ANPR systems allow for the speedy identification of cars implicated in illegal activity, assisting in investigations and improving public safety. These systems aid in the automated monitoring of traffic flow, congestion management, and the detection of traffic offenders in traffic management. ANPR technology is extremely important in toll collecting systems, speeding the payment process and reducing the need for manual involvement. ANPR systems also give useful data for traffic analysis, enabling data-driven decision making and infrastructure planning.

Several critical components are required for the creation of an ANPR system. The first phase is license plate localization, in which the system detects the region in an image or video frame that includes the license plate. Then, using character segmentation algorithms, individual characters are extracted from the license plate region. The characters are subsequently recognized and interpreted using optical character recognition (OCR) algorithms, which convert them into machine-readable text.

Building a successful ANPR system, on the other hand, is not without difficulties. License plate designs, typeface styles, lighting conditions, and camera angles all provide considerable challenges. The system must be strong enough to manage these differences while still delivering correct results in a variety of settings. Additionally, while building and deploying ANPR systems, ethical factors such as data protection and compliance with legal rules must be taken into mind.

Finally, the ANPR project intends to create a cutting-edge technology for automatic license plate recognition. The system aims to enable rapid and accurate recognition of license plates by using modern algorithms and machine learning approaches, contributing to increased public safety, streamlined traffic management, and enhanced data-driven decision making. This study will go into depth on the ANPR system, including the technique, implementation, obstacles encountered, ethical issues, and prospective future research and development paths.

Objective

The Automatic Number Plate Recognition (ANPR) project's goal is to create a robust and accurate ANPR system capable of automatically detecting and recognizing license plates from a variety of sources, such as pictures or video streams. The project's precise goals are as follows:

Develop methods and procedures for accurately locating and extracting license plates from pictures or video frames. This entails locating the area of interest (ROI) within the supplied input that contains the license plate.

Implement effective methods for segmenting specific characters from the retrieved license plate region. This process is intended to separate each character and prepare them for later identification.

Advanced OCR algorithms and machine learning models are used to detect and understand the segmented images. The goal is to turn the characters into machine-readable text.

System Accuracy and Efficiency: Achieve high accuracy and efficiency in license plate recognition. Even under difficult settings such as differences in license plate designs, font styles, lighting conditions, and camera angles, the system should produce consistent results.

Real-Time Processing: Design the ANPR system to recognize license plates in real-time or near real-time, allowing for speedy and efficient analysis of massive amounts of data.

Ethical Considerations: Address ethical issues related with ANPR technology, such as data privacy and legal compliance. Implement safeguards to secure personal information received by license plate recognition and to guarantee that the technology is used responsibly.

Performance Evaluation: Evaluate the ANPR system's performance using relevant criteria such as accuracy, precision, and speed.

Methodology

Pre-processing

Preprocessing the dataset is an essential step in preparing the data for machine learning tasks, such as image recognition or object detection.We collected the dataset from open source platform Kaggle and preprocessed the images which is the first and essential step. Each image in the dataset was subjected to the following preprocessing steps iteratively.

First, the image was resized to 256x256 so that all the images are of the same size. By resizing the images, we ensure that all samples have the same dimensions. It also helps to reduce the computational complexity and memory requirements during training.

Then, the resized image was converted to Grayscale . By converting the images to grayscale, the images are simplified by representing the pixels with single intensity values ranging from 0 to 255. It contains only brightness information and hence reduces the complexity of the dataset while retaining the important structural details.

Then the Gaussian blur is applied to this grayscale image. This technique is used to reduce noise in the image and also to smoothen out the irregularities in the image.

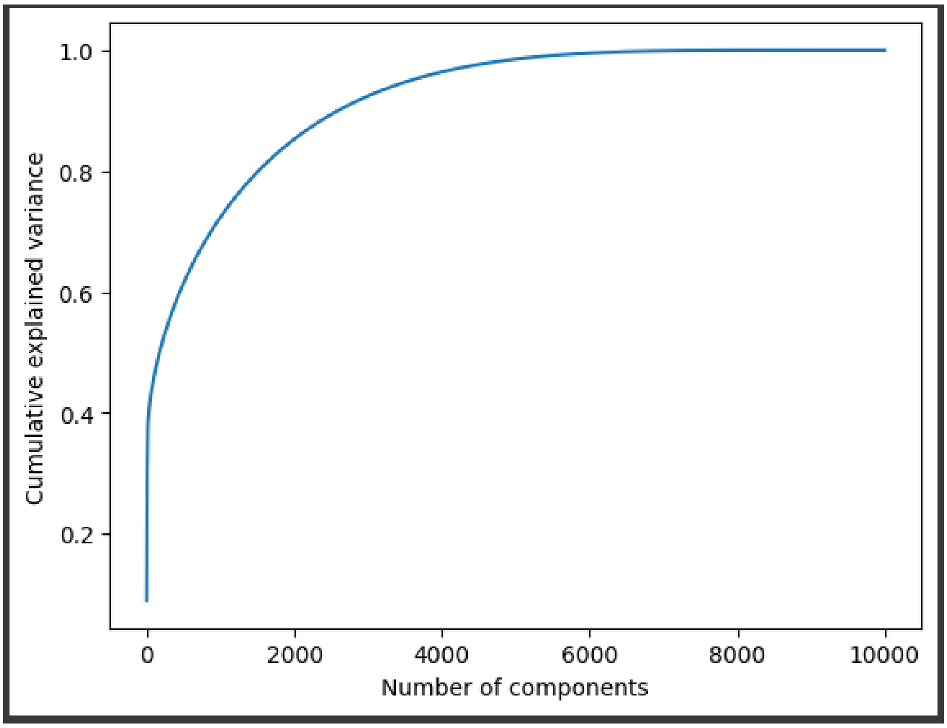
The final step in preprocessing the image is Edge detection. The goal of this step is to identify boundaries and sharp transitions in the image. We used the Canny edge detection method which highlights the edges.

Each of the images are preprocessed through the above steps and saved as preprocessed dataset. These preprocessing steps simplify the data, reduce noise, and extract important features, which can improve the performance of the model.

Analysis

Analysis is the process of inspecting, cleansing, transforming or modelling the data with the goal of discovering useful information. Here, we used the Principle Component Analysis (PCA) to reduce the dimension of the dataset. Dimensionality reduction aims at reducing the number of features or variables in a dataset while retaining the most important or the "Principle Components" in the data. This helps in training the model more faster and with lesser iterations.

First, to find the number of principle components, we created the scree plot which shows the principle components in the dataset. The x-axis shows the Number of Components and the y-axis shows the Cumulative variance. Our goal is to reduce the dimension of the dataset while retaining atleast 90% of the components which affect the training of the model. According to the scree plot, we need to take the number of components from where the curve stops increasing and starts to grow flat.



So, the number of Principle Components that retained atleast 90% of the main features were 4,500 components. This number was set and the PCA transform was performed on this number of components. The transformed data has the principle components in order which affects the model significantly. Hence, now we have reduced the size of the data while also retaining the important features that affect the model significantly.

Model Selection

After performing PCA on your dataset, the next step is to train a model for text recognition from the number plate images. One common approach is to use a Convolutional Neural Network (CNN) for this task. CNNs are well-suited for image recognition tasks. But since our dataset is unlabelled, we didn't have predefined labels for each number plate image. In this case, we won't be able to train a supervised model directly. So, using a Generative Adversarial Network which is an unsupervised learning model is a best option.

A GAN consists of two main components: a generator and a discriminator. The generator plays a crucial role in the GAN architecture, as it is responsible for generating synthetic data samples that resemble real ones. It takes random noise as input and transforms it into images. Through a series of learned transformations, the generator gradually improves its ability to create more convincing and realistic samples.

On the other hand, the discriminator acts as a binary classifier that evaluates the authenticity of the input images. It receives both real images from the training dataset and images generated by the generator. The discriminator's objective is to correctly discriminate between the real and generated samples. By continuously learning from the feedback provided by the discriminator, the generator adjusts its parameters to produce more authentic-looking images that are increasingly difficult for the discriminator to distinguish.

This adversarial dynamic leads to a training process in which both components improve iteratively, with the generator continually trying to produce more realistic samples, while the discriminator becomes more adept at distinguishing between real and generated images.

Training the Model

Once the PCA is done and the model is selected, the next step is to build and train the model. First, the generator model is created with using the Keras Tensorflow libraries. The initial layers of the generator are the input and the dense layers. Then we have a reshape layer which adjusts the dimensions of the input images. We have set the limit to be 10x10 with 128 channels. This then is followed by a convolution layer. This is where the image is synthesized. The last layer is again a convolve layer which is set to 1 filter so that the output has only one channel.

In the generator model, the choice of the number of filters in the first Conv2DTranspose layer (which is 64 in this case) is a design decision and can be adjusted based on the requirements and characteristics of the specific problem or dataset.

The number of filters determines the depth or complexity of the features that the generator can learn and generate. Increasing the number of filters allows the generator to capture more detailed and intricate patterns in the data. However, it also increases the model's capacity and the computational cost.

Choosing the optimal number of filters for the Conv2DTranspose layer requires experimentation and tuning. It depends on factors such as the complexity of the data, the desired level of detail in the generated images, and the available computational resources. Different values can be tried to find the balance between model complexity and performance.

Then the discriminator model is created. This model first has 2 convolve layers.

This is followed by a Flatten layer which converts the above 3D output from the previous layer into a one dimensional array. This process is required for the subsequent connected layers which is the dense layer. The dense layer is set to one output channel and a sigmoid activation function. This is where the classification occurs. The purpose of the discriminator is to distinguish between real and fake images, acting as the "adversary" in the adversarial training process of a GAN.

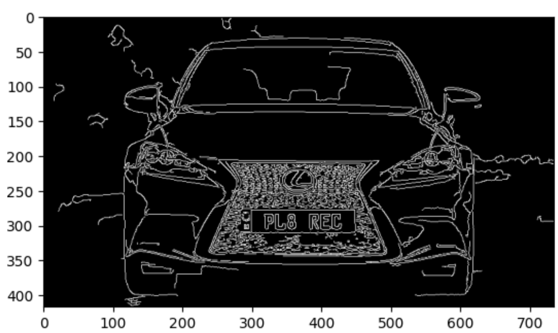
Once the models are built, we have to train them. To train the model, the first step is generating a random noise values which will serve as input to the generator model as it tries to generate similar image using random noise inputs. The generator uses this input and generates synthetic images. Then it selects a random sample from the preprocessed dataset. Since we used PCA, the model takes random image samples from this PCA variable. After this, the real and fake images are concatenated to create a combined set of images and then tables are created for the combined set of images indicating whether each image is real or fake.

Then the discriminator network is trained on the combined images and their respective labels. The goal of the discriminator is to learn to correctly classify real and fake images. The generator is trained via the GAN model using the new random noise and misleading targets. The objective is for the generator to learn to generate more realistic images that can fool the discriminator.

By iteratively training the discriminator and generator networks, the GAN model learns to generate increasingly realistic images, while the discriminator improves its ability to differentiate between real and fake images. This adversarial training process continues for the specified number of epochs (100 epochs), gradually refining the generator and discriminator to achieve better image generation quality.

Testing the Model

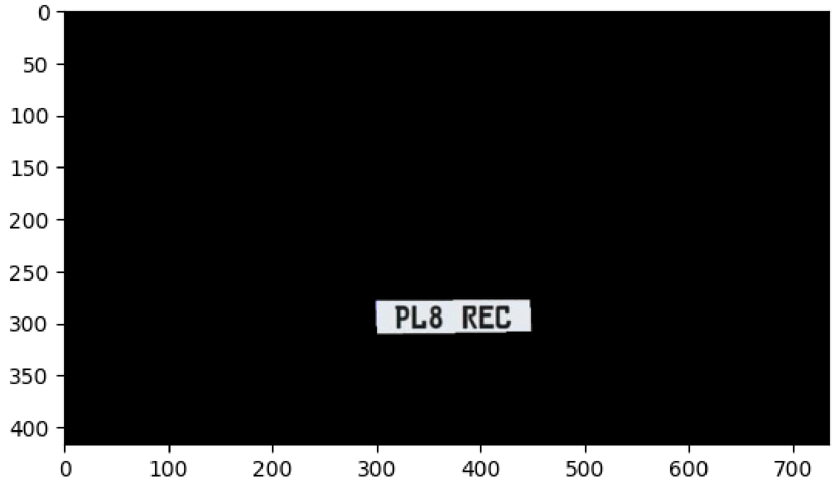
When testing the model the data needs to be pre-processed so that the model can identify features and generate similar images. For this process, the image data is first converted to GreyScale. After this, thresholding and Canny Edge detection is done. Canny edge detection is a technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed.

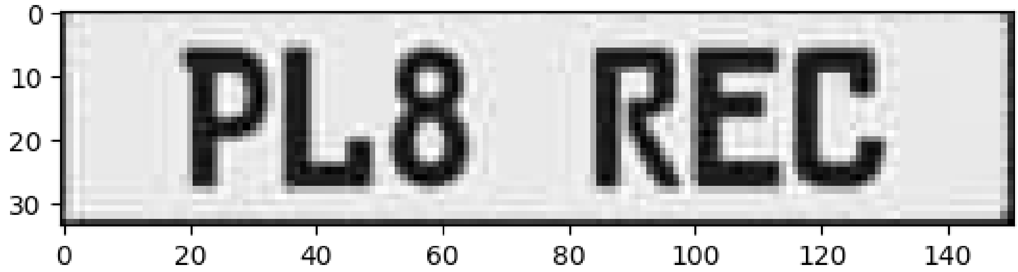


After the edge detection, the contour function was used. Contours are the continuous curves or boundaries that represent the shapes and objects present in an image. They are essentially the outlines of objects or regions with similar pixel intensity or color. The **cv2.findContours()** function in OpenCV is used to detect and extract contours from a binary image. It analyzes the input binary image and identifies the continuous curves representing object boundaries. It then returns a list of contours, where each contour is represented as a sequence of points (x, y coordinates) forming the contour curve.

Then, we iterate over the contours and get the one contour which has an approximate polygon shape and has exactly 4 vertices. If a contour with four vertices is found, the variable ‘**location**’ is assigned the value, storing the four vertices of the quadrilateral shape.

The value of location variable which is a 4x2 matrix, is then used to create a mask which is like a binary image with all 0s except the pixels within the vertices of location variable. This mask is then applied to the original image, which works like an AND function. This operation retains only the pixels in the original image where the corresponding mask pixels are non-zero. Essentially, it masks the original image with the contour region.





After Pre-processing the image, the image is sent to the model. The model now tries to generate an image which is similar to the original image. It works by passing the image through the generator model, which generates a synthetic image. The goal is for the generator to produce a realistic-looking image that resembles a license plate.

```
[([[16, 2], [70, 2], [70, 34], [16, 34]], 'PL8', 0.5099071860313416),
 ([[80, 2], [136, 2], [136, 34], [80, 34]], 'REC', 0.999786919245981)]
```

The output of the generator is an image, which resembles the license plate. The quality and accuracy of the output depends on the training of the GAN model and the quality of the training data.

After generating the synthetic image, this image is subjected to a character recognition system that extracts the text from image. The advantage of using a GAN (Generative Adversarial Network) model in the license plate recognition process is its ability to generate synthetic license plate images that closely resemble real license plates.

Evaluation & Improving the model

Effective evaluation allows us to assess the model's performance and identify areas for improvement. One common approach to evaluate a GAN model is to visually inspect the generated images. Visual assessment provides an intuitive understanding of the quality and realism of the generated samples. Human evaluators can examine the images for visual artifacts, blurriness, or any discrepancies from real images. Feedback from human evaluators can guide model improvements and help refine the training process.

To improve the GAN model, various strategies can be employed. One approach is to adjust the architecture of the generator and discriminator networks. Experimenting with different network architectures, layer configurations, and activation functions can lead to improved performance. Increasing the complexity of the model, such as adding more layers or nodes, can enhance its capacity to capture intricate patterns and generate more realistic images.

By iteratively evaluating, adjusting, and enhancing the model, we can achieve higher-quality of generated images and more accurate generation of complex patterns, leading to a more effective and robust GAN model.

Future Works

While our project achieved its objectives and produced satisfactory results, there are several avenues for future exploration and improvement. Here are some potential areas for further research and development:

- **Model Enhancement**: Investigate advanced GAN architectures, such as Conditional GANs or Progressive GANs, to further improve the quality and diversity of generated images. Explore architectural modifications, regularization techniques, or novel loss functions to address mode collapse or image artifacts.
- **Dataset Expansion**: Expand the dataset by incorporating additional sketch styles, diverse object categories, and more varied drawing techniques. This will enhance the model's ability to generalize and generate images in different artistic styles.
- **Conditional Generation**: Explore conditional GANs to enable control over the generated images. By conditioning the model on specific attributes or input conditions, we can generate images with desired characteristics, such as pose variations or specific object appearances.

Conclusion

This project's completion represents the pinnacle of our efforts to investigate Generative Adversarial Networks (GANs) for picture synthesis. We have learned useful insights and reached key milestones in the fields of artificial intelligence and computer vision along the way.

GANs have shown to be an extremely effective method for producing realistic pictures that closely mirror real-world data. We saw the creation of a sophisticated model capable of producing aesthetically attractive and diversified pictures by training a generator and discriminator network in an adversarial way. This achievement offers up a plethora of possibilities and uses in a variety of fields, including art, design, entertainment, and others.

The rigorous collecting and preparation of the dataset is one of the project's main highlights. We ensured that the GAN model learns from a rich source of information by selecting a broad and representative dataset of pictures. This results in more accurate and diversified image production. Preprocessing procedures such as resizing, normalization, and augmentation contributed to data standardization and improved the training process.

The GAN architecture's selection and design were critical to the project's success. We were able to achieve a delicate balance between inventiveness and realism by carefully developing the generator and discriminator networks. The generator network demonstrated its capacity to produce unique pictures, whilst the discriminator network efficiently learnt to distinguish between actual and fake images.

Training the GAN model needs a well-planned procedure. The training loop iteratively improved the model's performance by generating false pictures, combining them with actual photos, and optimizing the networks. The discriminator loss, for example, gave quantifiable measures of the model's development. The capacity of the model to catch fine features and provide cohesive outputs was validated further by visual evaluation of the produced pictures.

Finally, this experiment demonstrated the potential and capabilities of GANs in picture synthesis. The effective development of realistic and visually appealing pictures illustrates artificial intelligence's advancement. We are enthusiastic about the potential of generating even more intriguing and breakthrough applications in the future as we continue to explore further into the area of GANs. We are getting closer to realizing the full potential of machine-generated art and graphics with each step ahead.