Machine Learning Project Report
Image Cartooning

Tarunendra Malepati Roja Kamble Sushma Bobbishetty Sivaleela Gogineni

Abstract

Cartooning an image means transforming it into a cartoon form. Today, we can find a plethora of applications on the internet that will convert photographs to a cartoon look. Cartoon images often contain simple textures, crisp lines, and smooth colours, all of which show relevance for texture description. There are several facets to image processing. The picture of each element is seen as a 2-D matrix. Image processing, which includes object detection, object identification, object count, edge blurring, and other effects, is highly regarded in this field of research.

Introduction

Machine learning has spread into many sectors of the everyday life. we have decided to make a good understanding of the never-ending libraries of python. Our project "image-cartooning" involves various state of art libraries, which will make us become a proficient in the subject. Our project image cartooning does the conversion of the person's face into cartoon faces or other filtered faces. The method involves capturing the person or doing a live conversion of the face as done in the current social media applications like Snapchat, Google Meets, Zoom. The next step would be that face in image must be detected, and the boundaries of persons face must be calculated. Cartooning a digital image appears to be an interesting, enjoyable, and simple job to work on. We only need a bilateral filter and an edge detection algorithm to create an animation image from a digital image. These bilateral filters will help us reduce the image's colour or shading palette, which is a vital stage for the animation style, and edge detection will help us create a beautiful bold silhouette. For this process we need the machine to understand the persons face, so this machine must either be trained or must be using the trained class. There are many libraries that have a very advanced functioning such as Convolution Neural Networks, Residual Networks or pre trained models like You Only Look Once (YOLO v5). The advanced models that have the finest outputs and which are being used by most of the world, are using the process like Histogram of oriented gradients. By using these libraries, in the next process we can use various type of filter like cartoons images or face masks or also we can adjust the colour intensities. There are many tools in the market that perform the work, we want to perform the process that could make us understand the advanced libraries and make different kind of tests and want to get a best model with different kinds of filters. Cartoons are popular artistic styles like any other art. Cartoon effects are produced by using real-world photos. We use a set of images as training data. Additionally, our method trains far more quickly than the current model. Modern technology is increasingly present in our daily life. The main goal is to turn RGB into a realistic, cartoon image with a variety of filtrations or a blurred image with strong edge detection.

Motivation

The fact that we frequently encounter these many filters in daily life served as motivation for this effort. The way of thinking about it has always been fascinating. We aim to create a project that is appealing while also gaining a thorough understanding of image processing. This project sounds simple, but by working on it, we may test several types of image processing tools while also learning about all the image processing libraries. We can use a variety of classification techniques. We must use state-of-the-art techniques to achieve great accuracy, which offers us a thorough understanding of the most recent developments in machine learning.

Related Work

Cartooning image can be implemented using different existing tools that are available now. Python is the pool of libraries. For practical applications, it has a wide variety of libraries. **OpenCV** is one of these libraries. A cross-platform computer vision library is called OpenCV. Applications for recording and processing images and video are among them. The main applications for it include picture modification, object identification, face recognition, and a wide range of other astounding ones.
Other libraries that are available for cartooning are **Scikit-Image** (Active Contour), **SciPy** (Blurred images), **Pillow/PIL** (Image contrast), **NumPy** (Extract RGB channels), **Mahotas** – designed for bioimage informatics (Template matching for finding the wally), **Pgmagick** – GraphicsMagick (Resizing, Rotation, Sharpening, Gradient images, Drawing text, etc).

Cartoon effect has two specialties:

1. Highlighted Edges
2. Smooth colors

Most of the implementations available online w.r.t cartooning are OpenCV, CV2 and NumPy. Whereas we implemented using Dlib which results in a realistic cartoon image as the output.

As an extension of our project, using MediaPipe library in python can implement the following, face detection, iris, hands, pose, holistic, dace mesh, object detection like OpenCV, box tracking like OpenCV, hair segmentation, instant motion tracking, objectron, knift, etc.

The cartooning strategy makes use of a variety of image processing, computer vision, and deep learning methods to identify objects, abstract their characteristics, and then replace them with cartoon graphics and images.

Proposed Method

The methods that we are using in this project are, Dlib library and Pytorch library. In these library the methods that are being used are :

1. Histogram Oriented Gradients
2. Gaussian Blur
3. Bilateral Blur
4. Linear SVM
5. PyTorch

DLib

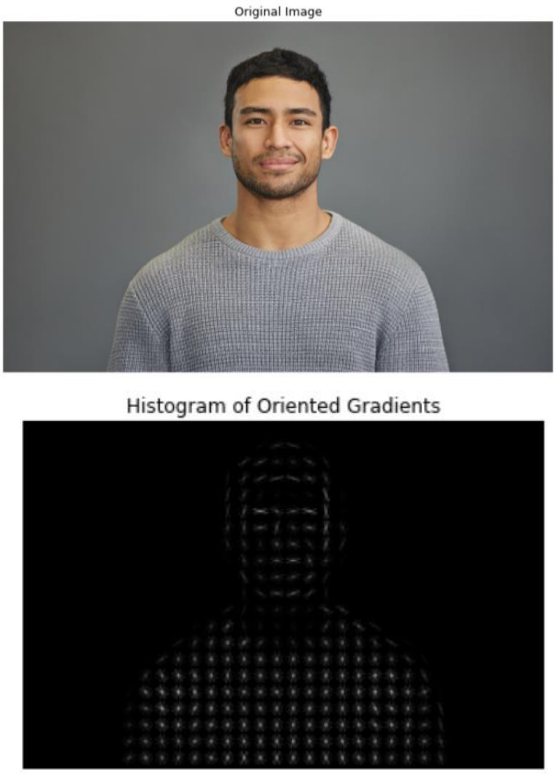
This is a C++ library; this library has various functionalities. We are using the frontal face detection methods in this library. This is a open source software. We have implemented this library in python.

The initial start of the library was from the year 2002, and since then it has expanded to include a wide variety of tools. It started to implement networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and a wide variety of other activities.

The library inputs the image, and it processes the image to its prebuilt model and it gives the features of the images. It gives the key points are the locations of the ears, eyes, nose, mouth etc. this model also the methods that could crop and align the image. The input is the PIL image, image is being read through the PIL Library, this will be taken into the method and the methods takes other argument such as the blur factor, crop, and alignment ratio. This is where the main project is needed as the facial features are to be modifies. This is the main methods. This is done using the NumPy array. This library uses the Histogram oriented gradients and the linear SVM model to make the predictions.

Histogram Oriented Gradients

HOG is essentially a feature descriptor used in both image processing and computer vision methods. This is a frontal face filter at the preprocessing stage. This helps in locating the face in image by 64128 patches as we have already mentioned. Obviously, the size of an image is not a restriction. Most of the time, patches of varying sizes are examined in many different locations across a picture. For the HOG feature descriptor calculation, we decided to use a patch that was 100 by 200 pixels. To begin the process of determining a HOG description, we must first determine the horizontal and vertical gradients. After all, the histogram of gradients is something that we want to figure out. This may be accomplished quickly and easily by filtering the image using the following kernels. The image is first filtered by utilizing Gradient Kernels matrices; only then is it possible to determine the horizontal and vertical gradients. The x-gradient triggers on lines that move in an ascending or descending direction, whereas the y-gradient triggers on lines that move in a left-to-right or right-to-left direction. When there is a sudden shift in the level of intensity, the size of a gradient fire will either increase or decrease. When the terrain is even, none of them will fire their weapons. Every single pixel in the gradient has both a size and a direction assigned to it. The largest gradient from any of the channels is used to determine the magnitude of the gradient at a pixel, and the angle of the largest gradient is used to determine the angle of all gradients. At this point, the image is divided into 88 cells, and a histogram of gradients is created for each of those 88 cells individually. Then we perform the 16*16 block normalization.



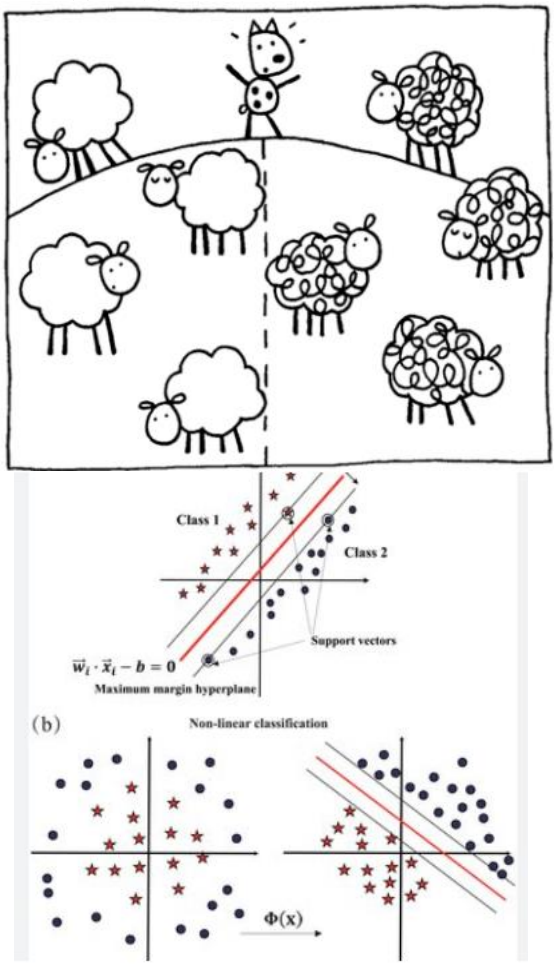
As you can see in the above picture the original image is a portrait of a man. In the histogram output we can see the direction of the edges that have been detected. At the corners we can see high magnitude. This makes a computer to understand the portrait. This is computer readable image to perform additional functions.

Linear SVM

Support Vector Machine, or SVM for short, is a type of technique used in machine learning that may be used to problems involving classification as well as regression. However, its primary function is to organize things into categories. When using the SVM technique, each piece of data is depicted as a point in an n-dimensional space. The value of each feature is determined by the value of a specific coordinate; hence the space is said to have n dimensions. After that, we give the data their proper classification by locating the hyper-plane that most effectively differentiates between the two groups.

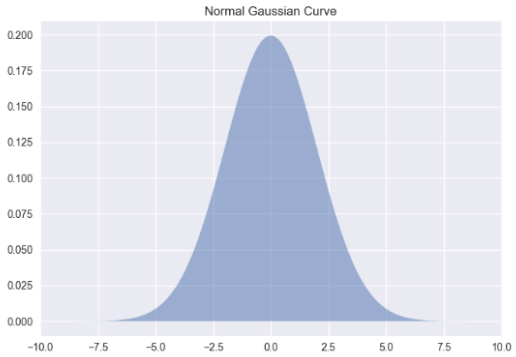
Illustration of the SVM approach in the form of a cartoon representation in the form of an example utilizing two discriminants, with the x and y axes serving as representations for each of the discriminants individually. (a) A method of organizing data in a linear fashion. The training data points are denoted by various symbols, with circles representing one class

and stars representing the other class. The support vector machine (SVM) attempts to determine the two ideal hyperplanes, which are illustrated as black lines in this illustration, that most successfully split the differentiating characteristics of the two classes (the maximum margin as labeled). This hyperplane can be found in the middle of the ideal hyperplanes. (b) The transformation of a nonlinear classification problem into a linear one by applying a function called, which is a chosen function that is related to the kernel function k in the following way: kxi,xj=xixj. This transformation takes place when a function called is applied. The following expression represents the classifier: f(x) = w(x) b.



Gaussian Blur

This is a method of removing lower freequenices from the input. This perform the action of a gaussian function. The gaussian function is like the intensities decreases with the distance from the centre with a factor . the curve for the gaussian blur is as shoen in the below figure.



The Gaussian blur when used on a plain images gives us a low frequency image which is blurred if the factor is being incresed. This is like reducing the pixels of the image in laymen terms. The basic of the gausian function is the foundation for the cartton image. The below is the image that has been gaussian blurred. This cannot be used to generate the output as the blur images makes the iamge to be unsatisfactory. We need to use a intelligence blurring. This is what we have performed in our project.

Bilateral Blurring

This is a non-linear filter in contrast to the gaussian filter. In this every pixel is being replaced by the weighted average of the previous pixel. Because of which the extreme noise is being compromised. This is one of the best usable filters in our case. In this cartooning if we give a weighted average of the pixel base do the area where it lies then the image will be a very satisfactory output. The bilateral if used without any predefined weights or by using a average weight function will be output sometime like the below figure.

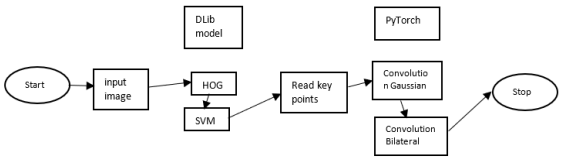
Pytorch Library

In this library there are many methods that can help perform actions on the images. In this we can convolve an image or we can even apply direct predefined methods to perform the actions. The method that we need to use are using a convolution of the filter with the image at the key points. This library has a wide library of the computer vision and natural language processing tools.

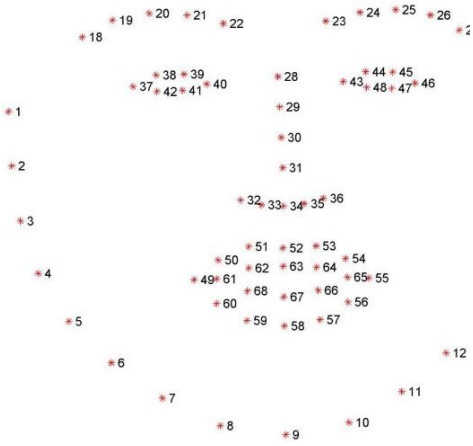
Experiments

Methodology

The method that is needed for the implementation of the project is that, first we need to build a model that can identify what kind of edit must be done or what is the touch up needed to the face for making it look like a cartoon. Not every part needs the same edit. When we first understood the process we then understood the steps on how to implement cartooning. So the flow chart will be as below:



As the above flow chart states the start for the model is the input image, then the input image is being processed in dlib in this library the model is going to the hog at the first in the hog the image is being processed and the output features are the gradient magnitude and the gradient direction, in the outline or the edges of the face portrait are being extracted. As we got the outline next, we need to point to the specific features and extract the exact data points. As the image can be zoomed or there might be various other objects in the background which look similar, we need to find the face. The common or similarity in a face is just the output so there must be a robust model that can predict the features to the best possible accuracy. For this the model needs to understand what is face is like. So, we need train the model with the enormous portraits. The portraits need to having the various tones of the faces at different exposures and at different brightness and at different angels. As said earlier the frontal face method is being called, in this method the hog is being built and the hog gets to see the outline of the edges and using the gradient angles we can predict the feature. The values or the array is being passed to the model that is built using the dataset, a linear SVM model is being used. As the dataset has the key points values. It is being a robust model. The model will be pretrained and now we are using the same model to test the input that is being given. Then the model gives us a array with key features. As shown in the figure below the datapoints are the first 17 points are of face boarders and next are right eyebrows etc.



```
plot_style = dict(marker='o',markersize=4,linestyle='-',lw=2)
pred_type = collections.namedtuple('prediction_type', ['slice', 'color'])
pred_types = {
    'face': pred_type(slice(0, 17), (0.682, 0.780, 0.909, 0.5)),
    'eyebrow1': pred_type(slice(17, 22), (1.0, 0.498, 0.055, 0.4)),
    'eyebrow2': pred_type(slice(22, 27), (1.0, 0.498, 0.055, 0.4)),
    'nose': pred_type(slice(27, 31), (0.345, 0.239, 0.443, 0.4)),
    'nostril': pred_type(slice(31, 36), (0.345, 0.239, 0.443, 0.4)),
    'eye1': pred_type(slice(36, 42), (0.596, 0.875, 0.541, 0.3)),
    'eye2': pred_type(slice(42, 48), (0.596, 0.875, 0.541, 0.3)),
    'lips': pred_type(slice(48, 60), (0.596, 0.875, 0.541, 0.3)),
    'teeth': pred_type(slice(60, 68), (0.596, 0.875, 0.541, 0.4))
}

plt.imshow(img)
plt.axis('off')
for face in landmarks:
    for pred_type in pred_types.values():
        plt.plot(face[pred_type.slice, 0],face[pred_type.slice, 1],
```

The points are being assigned to the pred_types in our code in the following manner. With this we can retrieve the specific labeled feature the we want. When we run the model and the points are being extracted the plot is being generated and that plots looks like

If we notice different colors for different points or features above that is because we have changed the color of the feature while assigning the values to the tuple. When plotted it considers the colors and plots accordingly.

The next steps after the feature extractions are the intelligent blurring. So when we see a cartoon it will have a high resolution image along with the cut edges. To make this we also need to make system understand that it should not blur near an edge and it should do the making blurring through the other regions. So, what we planned to do is the convolution of the filter in 2dimesnion. This will give the best output.

```
if align_corners:
    out = F.interpolate(out, half_size, mode="bilinear", align_corners=True)
else:
    out = F.interpolate(out, scale_factor=2, mode="gaussian", align_corners=False)
out = self.block_d(out)
```

As shown below we are using the pytorch library. In the library we are using the convolution and the interpolation methods. We use this model to get better model. We also add the pad the elements so that the uniformity is being maintained. Also, we need to set the portrait so we are using the crop and align function. In this function we will be using the basic NumPy array functions and setting the value.

Data set

The data set that we are using is that of a Kaggle this data set is as shown in the figure the key data points are to be classified and this data set has those points along with the possible values that those points contain. The values can be either oriented, exposed at various levels or the facial expressions might change. This data set has those all types of the feature points which makes it robust and because of which the model is giving a very high accuracy.

Results and Discussion

The result that we have achieved is the following the image

The result that we have achieved is a very satisfactory image. The image seems to be a painted portrait. When we compare the input image with the model generated is that the facial features of the person have been changed and the eyes are being enlarged in size. The shape of the nose is not changed but the view or the outline of the nose is being changed. The ear shapes are pulled out and the shape is changed to an extent that the face is recognized. Also the lips are enlarged and the color of the lips is being changed. The shape of the chin is changed the cheeks and chin is dragged so that the image looks more like a cartoon. If you see the above figure the overall shape is the same the definition of the edges is also huge but the main difference with the original image is the blurriness in the image this is because of the blurs filters that we using in the required places. If we see keenly the image has reduced the colors that are being used in the sense the original image if it has 7 colors it has been taken in 1 bin and an average color is being considered because of this the image looks more like a cartoon or hand paint. Also, the color is being uniformly distributed for the individual features. All of this has been achieved only because of using the HOG and understand the key features in image.

Data Analysis

The data analysis in our model is that the key features of the portrait. The points that are generated from the model are being used. These points have the shape or the value of that feature.

Conclusion

It is easier to scale the steps when we are aware of the context and the intended outcome. By conducting some research and learning the methods that are appropriate for the project, we must comprehend the procedure. Additionally, developing some profound insights and learning through each stage of the project that advances it will create the final product and a good result.

1. Potential future
2. Initial motivation
3. Previous work

Potential future

As an extension of our project, using MediaPipe library in python can implement the following, face detection, iris, hands, pose, holistic, dace mesh, object detection like OpenCV, box tracking like OpenCV, hair segmentation, instant motion tracking, objectron, knift, etc.

Scope:

1. we can build a mobile application or a web application.
2. we can make face masking(disguise) using histogram matching or other backgrounds editing filters.
3. we can build sentimental analysis by the face reactions and display the emotion.
4. We can make gender or age analysis.

Initial motivation

- The fact that we frequently encounter these many filters in daily life served as motivation for this effort.
- The way of thinking about it has always been fascinating. We aim to create a project that is appealing while also gaining a thorough understanding of image processing.
- This project sounds simple, but by working on it, we may test several types of image processing tools while also learning about the image processing libraries.
- A variety of classification techniques is available. We need to apply state-of-the-art technique, which gives us a comprehensive understanding of the most recent advancements in machine learning in terms of application and precision.

Previous work

Cartoon conversion using the following filters of CV2 library.

Gaussian

A Gaussian filter is a low pass filter used to blur certain areas of a picture and reduce noise (high frequency components).

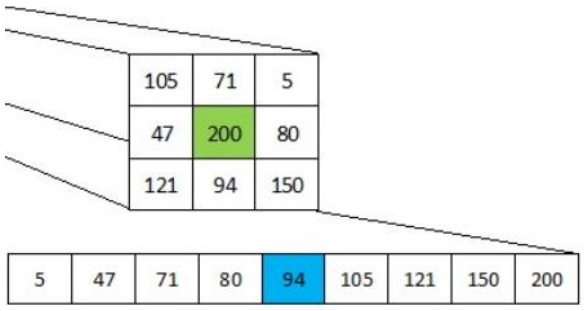
1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Bilaterial

In Bilateral blur filter the intensity of the each is changed with weighted average. This is like gaussian blur. But this is a nonlinear, edge smoothing filter. The function is multiplied with each pixel and the convoluted value is stored as the output.

Median

This as the name suggests does the median of the surrounding pixels and store that as the output. This is a best example of the non-linear filter. This helps in reducing black and white noise.



Acknowledgement

We are very thankful for the university to provide us the resources that were needed during the project work. We also acknowledge our professor Yuan Jing who taught the course and made us better understand. Professor has been very helpful through the project and helped us with the depth knowledge. Also the supporting Staff Arefa Patwary and Dang Nghia. we are very thankful for each and every individual for making our project successful.

References

[1]https://data-notes.co/snapchats-filters-how-computer-vision-recognizes-your-face-9907d6904b91

[2]https://nykdaily.com/2020/11/types-of-cartoon-drawings-comic-strip-gag-editorial-animated-illustrative/

[3]https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

[4]https://www.ibm.com/cloud/learn/convolutional-neural-networks

[5]https://medium.com/sciforce/face-detection-explained-state-of-the-art-methods-and-best-tools-f730fca16294

Contributions

Tarunendra Malepati – Research
Programming
Presentation
Documentation

Roja Kamble - Research
Programming
Presentation
Documentation

Sushma Bobbishetty - Report Documentation
Research

Sivaleela Gogineni - Report Documentation
Research