COMPUTATIONAL PHOTOGRAPHY

Project 3 Report

JUNE 5, 2020

MATTESINI RICCARDO – BUNCHALIT EUA-ARPORN
2019280795 - 2019280794

Preliminaries

Our research topics are different. Riccardo's research is on Multi-Agent reinforcement learning and game theory. Bunchalit's research focuses on Time-series forecasting model.

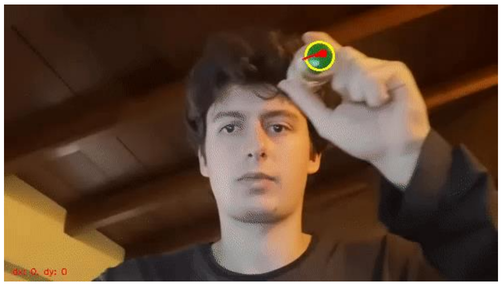


Figure 1 Object tracking - Frame from video

First of all, we tried to implement an object tracking algorithm, which simply works by taking the range of colors (0-255,0-255,0-255) of the object that we want to track, and on each frame of the video it draws a circle around the area with that color, by means of a mask (using cv2.inrange() to detect the area). We then decided to test something more articulate. We could of course continue to develop the object tracking idea, but we realized very soon how training a reinforcement learning algorithm on some video, requires way too much computational power. Then, we redirected and tried to implement algorithms that are more practical to our limitation.

Therefore, this project is structured into two parts, first one more related to Riccardo's research area while second is closer to Bunchalit's one.

Introduction

Our setup is a Lenovo y50-70, 5th generation intel CPU, gtx860m GPU and 16Gb of ram. To capture the videos and pictures, for both object tracking and 3d reconstruction, a Huawei p20 pro was used. The program is run on a virtual machine with 3gb of ram. With this setup, even for a simple video, we couldn't accept to wait maybe 2 days or more to get the result of our algorithm and meanwhile wasting that amount of time each time our algorithm didn't give a proper result. So, we decided that working on pictures would have been smarter and faster.

Contribution

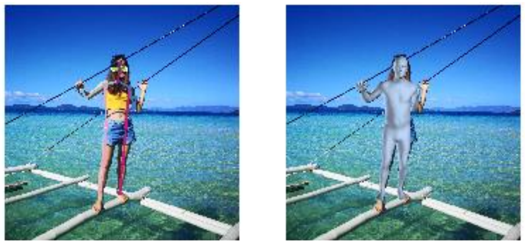


Figure 2 Output shape for female model

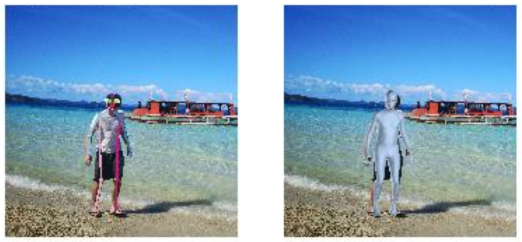


Figure 3 Output shape for male model

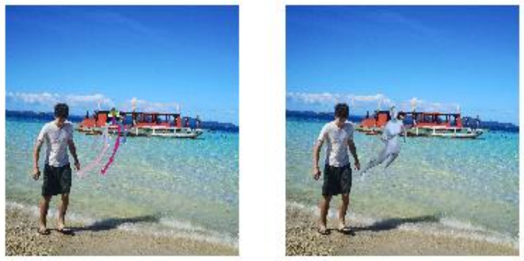


Figure 4 Output shape for male model

As we mentioned above, the project takes into account pictures rather than videos. What we want to achieve, is to get a 3d representation of the subject appearing in an image. There are many works that address this problem, because it is a good basis for a wide range of other implementations [1] [2] [3] [4]. The main problem is to transfer 2d data into a 3d model. We make use of SMPL, a realistic 3D model of the human body that is based on skinning and blend shapes and is learned from thousands of 3D body scans. The testing pictures used are taken from our past holidays pics, with strong light. As we can see from the output images on the right, in the case of the female model, the reconstructed 3d output is of reasonable quality. For the male case, the output is not completely wrong (Fig.3), but also not perfect(Fig.4). This is due to the fact that we are looking for objects in the center of the image. In this case, the generative adversarial network, that has to tell whether the object we are dealing with is a person or no, is given a person on the left but which is supposed to be in the center, for which the output is a ghost style figure. This approach was preferred, in order to avoid a heavy implementation consisting of 2d-to-3d paired data. Another case is the high contrast one, where the object looks black, and the output's an unnatural shape. We can overcome this problem by training the adversarial network in a more complete dataset or discard not centered images. A better solution is to use object recognition as in the preliminaries (Fig.1) to find the object of interest and then move it to the center of the image, even by cutting the sides of the image, which are not interesting to us (Fig.4 to Fig.3).

Conclusion

As was mentioned, this implementation leads to many other possible ideas. We can for example take a video, extract each frame and extract the 3d mesh from each frame. Finally, putting all the frames together we can get a 3d video representation of the subject moving. In this case, we don't deal only with a simple 2d input image, but other features, such as distance, distortion, position of hidden features, etc. can be extracted from different frames and then collected into a unique output. Another possibility is to use multi agent reinforcement learning to get an output with different subjects in the same picture/video. The computational weight of this implementation would be heavy though.

In the second part, mainly referred to Bunchalit's research area, we want to encode time-series data as images by using Gramian Angular Fields(GAF) technique.

Introduction

In computer vision, Convolutional Neural Networks (CNN) have been studied extensively and have seen a lot of successful applications as presented in Deep Learning for Image/Video Analysis talk from GUO Yuchen. Motivated by this we then tried to find a way to apply this technique to our research area. We found very interesting papers [5] and [6]. In the papers, they applied a Gramian Angular Fields(GAF) encoding technique to a time-series data. By encoding a time-series data as images, researchers then leverage Convolutional Neural Networks (CNN) technique to classify the encoded images of time-series data. The encoding basically maps a data from n $\longmapsto$n² in a form of Gramian matrix. Gramian Matrix of a set of n vectors is a matrix defined by the dot-product of every couple of vectors. The step to encode the time series is

1) Given a time series $X = \{x_1, x_2, \dots, x_n\}$ we can scale it onto [-1,1] using

$$\frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) + \min(X)}$$

2) We can represent the data in a polar form (r, ϕ) where $\begin{cases} \phi = \arccos(\tilde{x_\iota}) \\ r = \frac{t_i}{N} \end{cases}$ noted that normalized $\tilde{x_\iota}$ lies in [-1,1], t_i is the time stamp and N is a constant factor to regularize the span of the polar coordinate system.

3) Then we exploit the angular perspective by considering the trigonometric sum between each point and get

$$G = \begin{Bmatrix} \cos(\emptyset_1 + \emptyset_1) & \cos(\emptyset_1 + \emptyset_2) & \dots & \cos(\emptyset_1 + \emptyset_n) \\ \cos(\emptyset_2 + \emptyset_1) & \cos(\emptyset_2 + \emptyset_2) & \dots & \cos(\emptyset_2 + \emptyset_n) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\emptyset_n + \emptyset_1) & \cos(\emptyset_n + \emptyset_2) & \dots & \cos(\emptyset_n + \emptyset_n) \end{Bmatrix}$$

$$= \tilde{X}^T \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}^T \cdot \sqrt{I - \tilde{X}^2}$$

If we let $< x, y > = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ then

$$G = \begin{Bmatrix} < x_1, x_1 > & < x_1, x_2 > & \dots & < x_1, x_n > \\ < x_2, x_1 > & < x_2, x_2 > & \dots & < x_2, x_n > \\ \vdots & \vdots & \ddots & \vdots \\ < x_n, x_1 > & < x_n, x_2 > & \dots & < x_n, x_n > \end{Bmatrix}$$

which is then our Gramian Angular Fields (GAF) images.

Contribution

Due to time limit, we can only finish the encoding part but not the Convolutional Neural Networks classification part. However, we have implemented to the real stock market time-series data. We selected 4 stocks from 2 different industry. Which are Google and Microsoft, United Airline and Delta airline. We want to see whether these encoded images can be intrepid with human's eyes. The results of our implementation are shown in Table1.

Conclusion

Even without CNN classification, but we can see that the encoding is somehow reflex the actual characteristic of the data. Data from technology industry have a dark area in the top left of the images while for airlines, the dark area is in a lower right of the image.

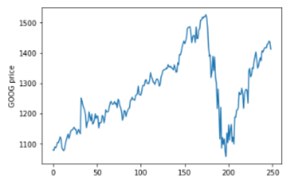
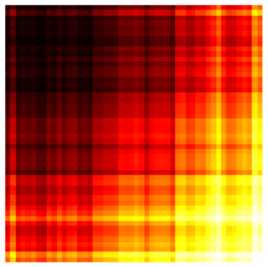
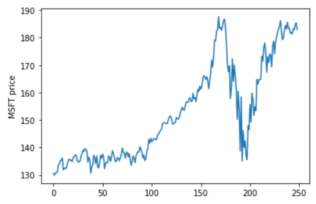
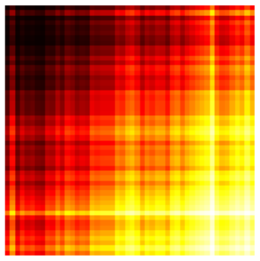
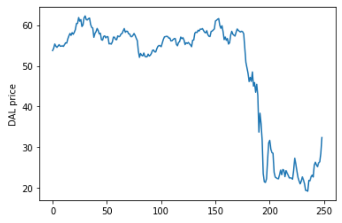
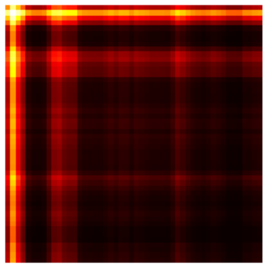
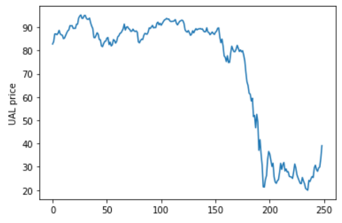
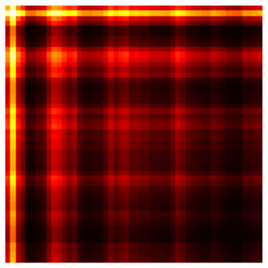
Company stock	Actual data graph	Encoded imgae
Google		
Microsoft		
Delta Airline		
United Airline		

Table 1: Examples of time-series encoded data

Bibliography

[1] H. Yu and al., *Self-supervised Learning of Motion Capture,* 2017.

[2] K. Angjoo and al., *End-to-end Recovery of Human Shape and Pose,* 2018.

[3] O. Mohamed and al., *Neural Body Fitting: Unifying Deep Learning and Model Based Human Poseand Shape Estimation,* 2018.

[4] Z. LuYang and al., *Learning to Estimate 3D Human Pose and Shape from a Single Color Image,* 2018.

[5] J.-H. C. and Y.-C. T., *ENCODING CANDLE STICKS AS IMAGES FOR PATTERN CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS,* 2020.

[6] Z. W. and T. O., *Imaging Time-Series to Improve Classification and Imputation,* 2015.