**FIT5221**

**Assignment 3**

**Collage Key Frames**

**Prepared by:**

Audi Rivai – 30399262

Jianpeng Fu – 27462870

Master of Data Science - C6004

Faculty of Information Technology

Monash University

1. **Introduction**

Have you ever wondering how social media have different kind of features when it comes to picture? For example, Instagram's newest app is a better way to make collages of a picture that can create multiple images in on frame photos. Creating a photo collage is one of the examples of image processing. But what if we have a lot of picture or videos frame and create a photo collage. How do we choose the key frames? Which frame is the best suit for the collage? This type of question we will discuss more by using deep learning method called Faster RCNN and SSD. We will also extract the feature using HOG (Histogram of Oriented Gradients) feature extraction.

1. **Faster RCNN and SSD**

In the object detection, there are several methods that can be use. In this task, we will be looking some of the method which are Faster RCNN and SSD. The foremost broadly utilized state of the craftsmanship adaptation of the R-CNN family — Faster R-CNN was to begin with distributed in 2015. This article, the third and last one of a arrangement to get it the essentials of current day object detection elaborates the technical details of the Faster R-CNN detection pipeline (Ananth, 2019).

As for SSD, is designed for object detection in real-time Speedier R-CNN uses a region proposition network to make boundary boxes and utilizes those boxes to classify objects. Whereas it is considered the start-of-the-art in accuracy, the entire prepare runs at 7 frames per second. Distant underneath what a real-time processing requires. SSD speeds up the method by dispensing with the require of the locale proposition arrange. (Hui, 2018). We will use this to detect the object and extract the feature.

1. **HOG (Histogram of Oriented Gradients)**

The HOG (Histogram of Oriented Gradients) may be a feature descriptor utilized in computer vision for picture processing for the reason of object discovery. HOG features are extricated from expansive numbers of facial images to be utilized as portion of the recognition mechanism (Ahmad, 2019). As we know all the method to use, we will use this image to create a photo collage.

1. **Instruction to run code**

Please upload the whole folder ‘FIT5221\_ass3\_30399262\_27462870’ to your Google Drive under your ‘My Drive’ (which is also the main page of your Google Drive). It is important because the ROOT in our code has been fixed and cannot be modified on your side. Then open ‘assignment3.ipynb’ using Google Colab.

1. **Extract Keyframes**

Before talking about frame extraction, we would like to give a definition to ‘**Key frames**’ here. Key frames can be understood as important frames that is a shortened version of video, they still have content relationship in between. But that is not ‘key’ enough, for example, if the input video is about a birthday party, then a bunch of frames showing smiling faces is not expected, we only want the frames that can show it is a birthday party and the smaller number of frames the better. So, in our extracted key frames, it is less likely to have many frames for one scene/background.

In previous section we already discuss the method will be use. To get the keyframes from the video, we will need to do several steps:

* Extract all the frames from the video using FFMPEG/OpenCV
* Turn all the frames into a grayscale image so that we can find the similarity by smoothing the line graph and finding the local maximum as the key frame
* Use pre-trained model object detection model which is SSD or Faster-RCNN from TensorFlow to get the boxes
* Extract the feature of frames using the HOG feature extraction.
* Find the similarity of the frame’s features based on the threshold and using SSIM (Structural similarity index measure) to get the rest of the key frames.

We will discuss the step for extracting the frame.

* + 1. **Extract all the frames**

In this task we will be using some example videos given by our lecturer. Frames are extracted from videos by using FFMPEG and OpenCV (Audi uses FFMPEG and Jianpeng uses OpenCV). This will get all the frame based on the duration of the video below will be the example of the frames.





Figure 1. Example frames from the video using FFMPEG/OpenCV

The result shows that there are around 100 of frames from 5 second duration video. Next, we will show how to choose which frame is the important for the collage.

* + 1. **Find the Local Maximum**

Once we already have the extracted frames, change them into grayscale then we will calculate the average pixel intensity difference of each frame and the previous frame. For example, in the Figure 1, we can see that frame 1 and frame 2 are a bit similar, next we calculate between frame 2 and frame 3 and so on. We will calculate all the current and the previous frames to know the similarity (in a color level since we are using pixel values). We think it is a reasonable method because normally when there is a cut to another scene, the color will change significantly.

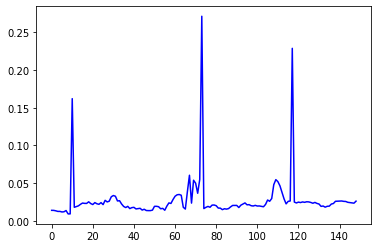
****

Figure 2. Difference Result between current frames and previous

From the line chart we can tell that there is high error value, it means that there are many differences between current frame and previous frame. Once we got the result, we will smooth the line so that we can have the local maximum. One important thing here, we tried using a threshold to cut the error values which will give 3 peak points in this case, but the threshold is not consistent for all videos. Even using percentile of error cannot help with the performance. So, we decided to find local maximum.

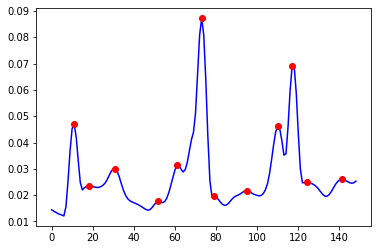


Figure 3. Smooth line and Local maximum results

We can see that there are number of dots. These dots are the local maximum or called the important frames. However, some frames that we pick from the local maximum still have similar picture but the number of frames has been significantly reduce from 120 to 12. Therefore, in the next section we will be using pre-trained model object detection model which is SSD.

* + 1. **Keyframe Selection and Feature Extraction**

From the previous task, using this video we already get 12 different frames. We use these frames to perform some object detection using SSD and find 10 objects from each frame in total there are 120 objects since there are 12 frames. Since there is a run time constraint, we decided to use SSD instead of faster R-CNN.



Figure 4. 2 Samples Object from Frame 1

After we get the object in every frame, we use HOG to extract the feature for every object in each frame. Below will be the step for the HOG feature extraction.

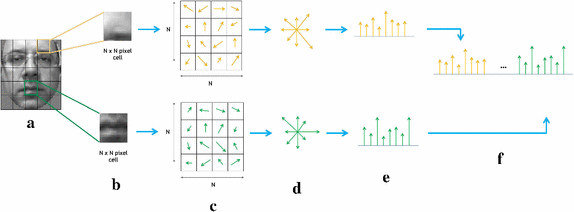


Figure 5. HOG Feature Extraction sample

We can see from the steps; image is separated in cells of estimate N × N pixels. The orientation of all pixels is computed and gathered in an M-bins histogram of orientations. At long last, all cell histograms are concatenated to build the final features vector (Carcagnì, 2015). In this video, we will have 288 features that have been extract (10 boxes). Since there are some frames that have almost similar with other frames, we need to reduce the number of frames by filtering the HOG feature based on the threshold on HOG distance.

We find that even after filtering with HOG distance, there are still similar frames in our list. In order to achieve our defined ‘key frames’, we apply SSIM to the rest frames in the list. SSIM filters the frames by looking at them from a structure level.

1. **Create Photo Collage**

In previous part we manage to extract the keyframe, now we need to put the keyframes into on photo collage. We will demonstrate the step.

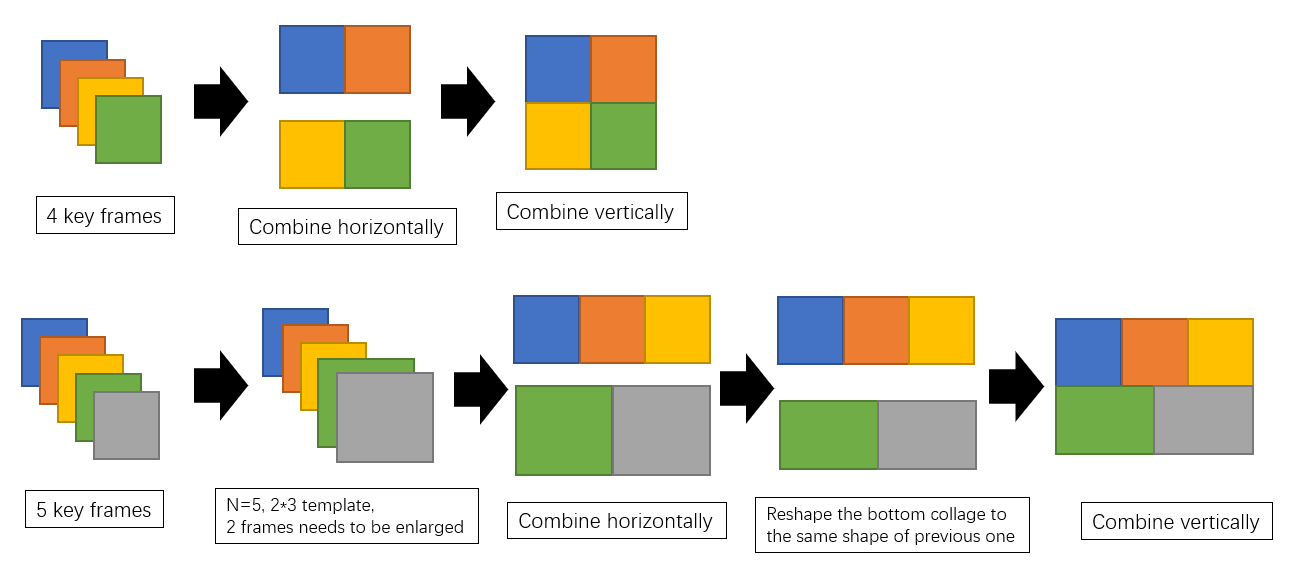


Figure 6. Flowchart representing our collage function

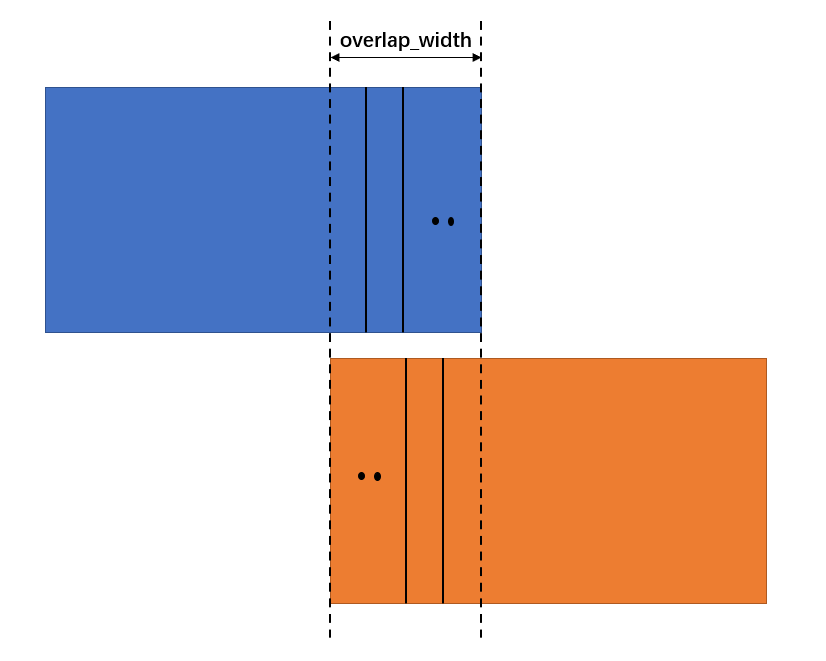
For the collage function, we have defined 4 functions in python, ‘**find\_factor**’, ‘**find\_closest\_factor**’, ‘**overlapping**’ and ‘**collage**’.

In this project, the number of key frames is unknown, therefore the arrangement of these frames in the collage is very important. At the beginning, we applied ‘find\_factor’ function to help with finding the closest factors ‘a’ and ‘b’ that a\*b=n. But one problem here is that if n is a prime number, then a=1, b=n, and the output shape of collage is not acceptable. Then we constraint the maximum difference between ‘a’ and ‘b’ to be 1, and we are going to resize some of the frames to fulfill the collage. For example, if n is 5, then the factors will be 2 and 3 with an extra string output ‘enlarge’ illustrating that the size of some frames should be enlarged.

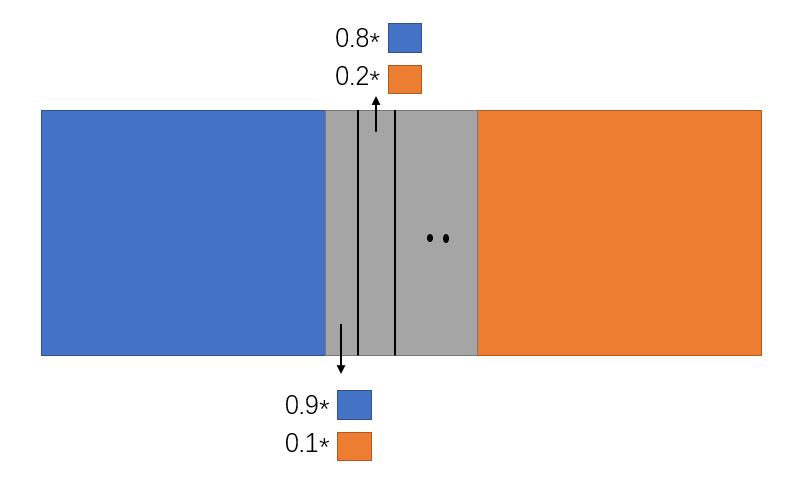
The overlapping function helps to combine two images with a blurred connection. There are 6 inputs to this function:

* + 2 input images
  + Direction, which is horizontal or vertical
  + Overlap weight is the area of overlapping
  + Slice, is the number of slices in the overlap area
  + Frame size is the original frame size.

As shown in the flowchart of collage function, we need to combine images either horizontally or vertically, and this direction tells which part of pixels should be used in the following process. For example, if it is horizontal, then the right part of the first image and the left part of the second image will be used.



(a)



(b)

Figure 7. (a) combining Image. (b) Shading Image Process

Then, for each slice in the overlap area, we apply a shading weight which is from 1 to 0 step by step in first image and other way round in second image. The image in figure 7(b) is an example.

The number of slices strongly relates to the effect of shading in the output image, the more the slices, the better the transition between two images. It is always better if we can set weights to each column/row in the overlap area, but it is time-consuming.

In the collage function, we called the “**find\_closest\_factor**” function to give the arrangement of frames in the collage, and decide which frames need to be resized. Then, calling the overlapping function to combine a new frame with the previous outputs.

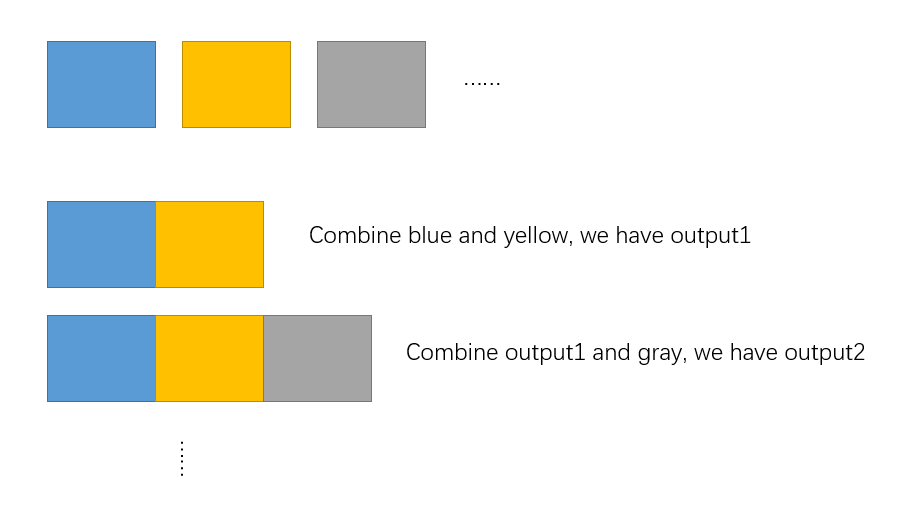


Figure 8. Final process of create photo collage

Lastly, combine all horizontally combined outputs vertically to get the final collage output. Figure 9 shows a sample output using 5 test images.

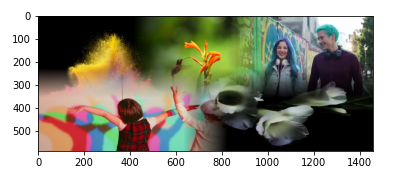


Figure 9. Sample output of collage function

1. **GIF Frames**

This part similar to what we did in removing sharp boundaries in collage function. Now the weights will be applied to the whole image instead of a part of image. And we can clearly see the transition in output frames.

Also in the ‘**gif\_frame\_making**’ function, a folder called gif\_frame\_folder will be created automatically which stores all the frames for GIF (with transition).

1. **Summary and Result**

Figure 10. Process of Extract Keyframe

Using this process, we are now allowing the code to extract the keyframe and create the photo collage. First filter frames from a color level, then using HOG distance to remove frames from a feature level. Lastly, using SSIM to select key frames from a structure level. The table below stores all outputs.

|  |  |  |  |
| --- | --- | --- | --- |
| Video Name | Keyframe Number | Result Photo Collage | Time Consuming |
| 178\_15  .mp4 | 3 |  | 42.2 second |
| 84\_19  .mp4 | 2 |  | 53.1 second |
| 135\_14  .mp4 | 4 |  | 1 minute 12 second |
| 177\_1  .mp4 | 3 |  | 1 minute 2 second |

Table 1. Result of the Photo Collage

We can see from the table; every video returns a different keyframe and create a different photo collage. For the photo collage, there are some improvement that can be made. We can use some clustering method to extract the keyframe. However. This will take a long time to process because they need to find the right cluster point to solve it. As a summary, deep learning and computer vision can create a unique and creative art. There are many ways to create an outstanding picture or frames to show people how deep learning can do.

1. **Contribution**

Audi: Frame extraction using FFMPEG. Key frame extraction using clustering (although it is not shown in this report), HOG distance and SSIM, report writing.

Jianpeng: Frame extraction using OpenCV, collage function, GIF frame making function, key frame extraction.

All work is equally distributed and both of us are satisfied with the working process.

# References

Ahmad, R. (2019). *A Take on H.O.G Feature Descriptor*. Retrieved from Analytics Vidhya: https://medium.com/analytics-vidhya/a-take-on-h-o-g-feature-descriptor-e839ebba1e52

Ananth, S. (2019). *Faster R-CNN for object detection*. Retrieved from Towards Data Science: https://towardsdatascience.com/faster-r-cnn-for-object-detection-a-technical-summary-474c5b857b46#:~:text=The%20most%20widely%20used%20state,was%20first%20published%20in%202015.&text=In%20the%20rest%20of%20the,CNN%20as%20a%20detector%20network.

Carcagnì, P. a. (2015). Facial expression recognition and histograms of oriented gradients: a comprehensive study. *Springer Plus*, 645. doi:10.1186/s40064-015-1427-3

Hui, J. (2018). *SSD object detection: Single Shot MultiBox Detector for real-time processing*. Retrieved from Medium: https://medium.com/@jonathan\_hui/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06