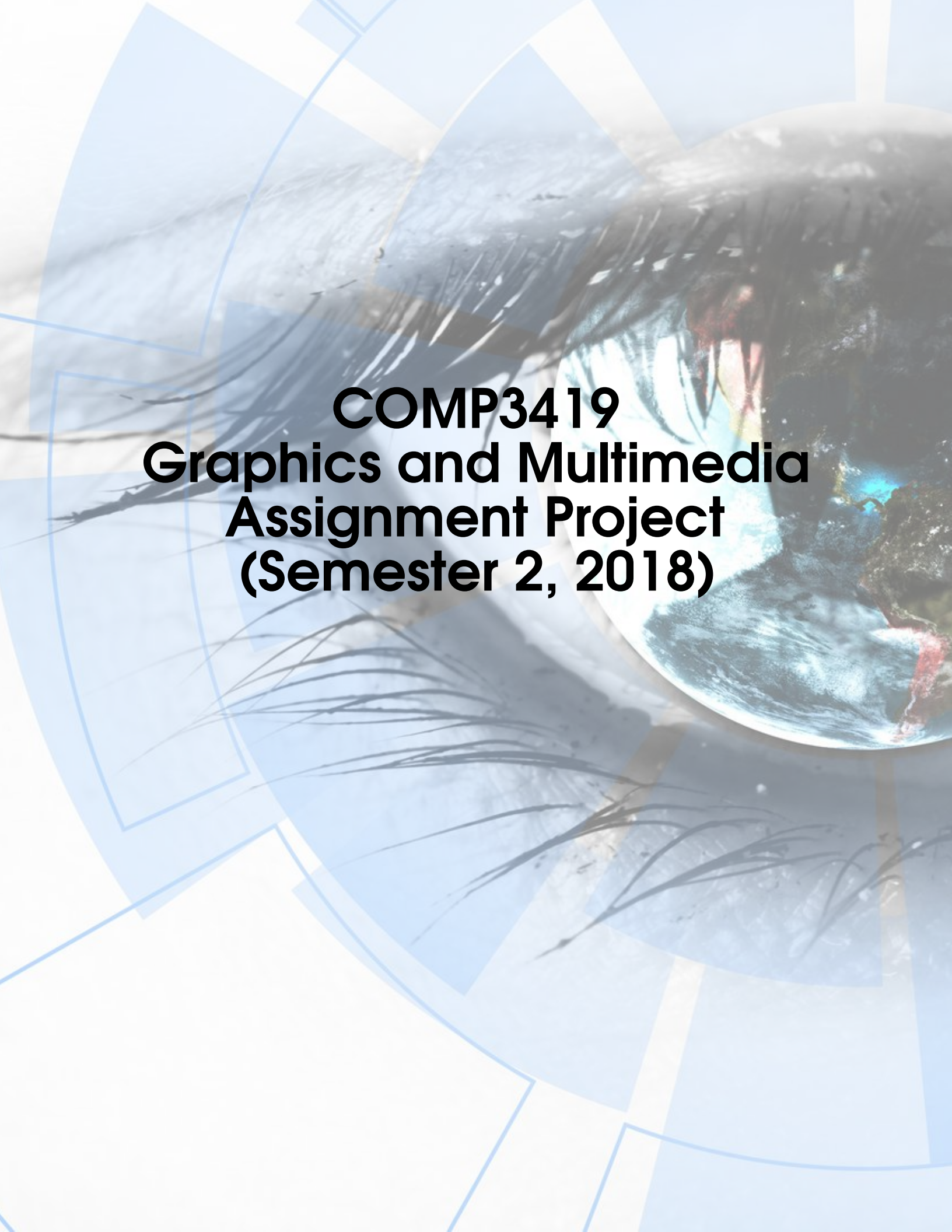
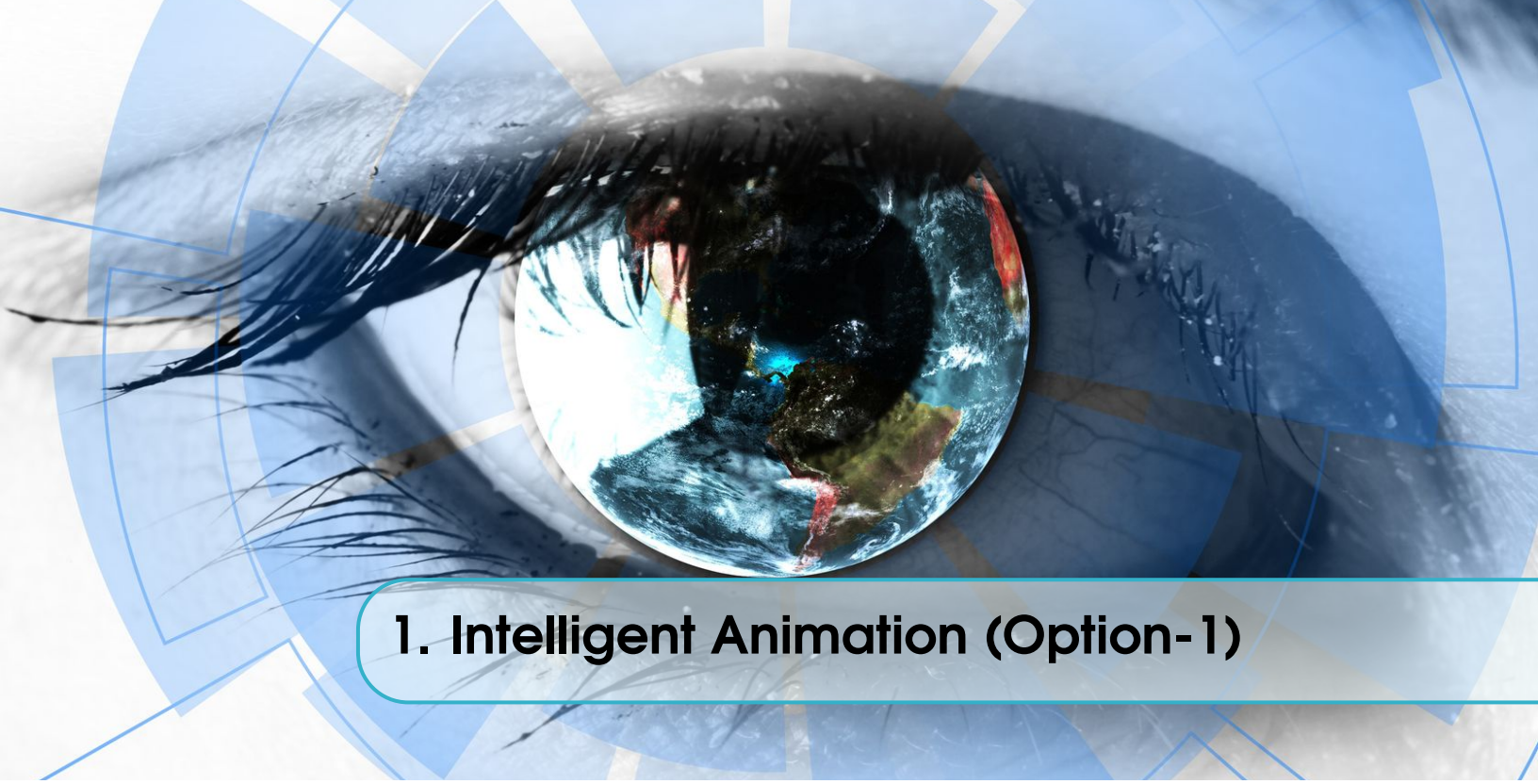


COMP3419
Graphics and Multimedia
Assignment Project
(Semester 2, 2018)



1. Intelligent Animation (Option-1)

1.1 General

This is an individual assignment (Option-1) and worth 25% of the total assessment of this unit of study. In this assignment, you are required to program a short video involving digital video processing, compositing and 2D animation techniques. The output video is a piece of animation based on a provided video clip. You are welcome to use ANY programming languages to complete your assignment.

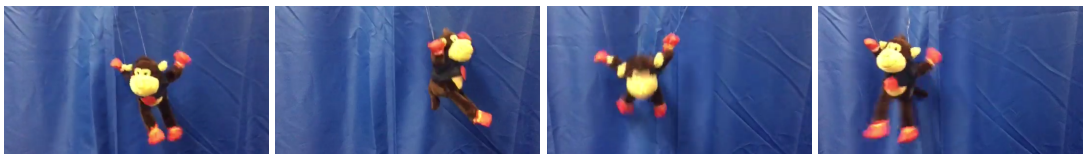


Figure 1.1: Some example scenes of the input video.

1.2 Main Objectives

1.2.1 Motion Capture

[5%] The body of a monkey is labelled with red markers. Segment the red markers/the monkey and use the coordinates of them to track the body motions. Some morphological operations might be needed to enhance the segmentation of the red markers. Alternative strategies can also be used for motion capturing, e.g., optical flow, boundary detection, etc. Design a data structure to represent the sequence of the captured body motions.

1.2.2 Replace Background and Marionette

[7.5%] Replace the blue background with your own dynamic background which can be programmed animations or a video. Render your own character to replace the moving monkey according to the captured motions in a new video. You should use the animation techniques (will be introduced in

labs) to achieve this and simulate the gestures of the monkey as much as you could. The replaced character should have at least five connected components, including a body, two arms and two legs.

1.2.3 Intelligent Objects

[5%] Add at least two types of randomly moving objects to your video to interact with the moving marionette in two different ways (e.g. collision, tracking, etc.). The interactions should be affected by the motions of both the added objects and the marionette. [1] Trigger special effects when interactions happen using image processing techniques. More intelligent objects are encouraged. The marking will depend on the design of intelligent objects and their interactions.

1.2.4 Sound Track

[2.5%] Program at least two sound tracks for your video. The sound tracks should be related to the interactions between the moving objects.

1.2.5 Technical Report

[5%] Draft a 4-6 page technical report to demonstrate your pipeline. The main sections of the report should at least include Introduction, Implementation and Conclusion. The implementation can discuss the algorithm and your experimental results. Your report should be written following the scientific style and formatted with LaTeX. Do not worry if you have not used latex before. It has a similar syntax to HTML. You can find a handy online latex editor at `https://www.sharelatex.com/`. An alternative choice is overleaf which can be found at `https://www.overleaf.com/`.

1.3 Constraints

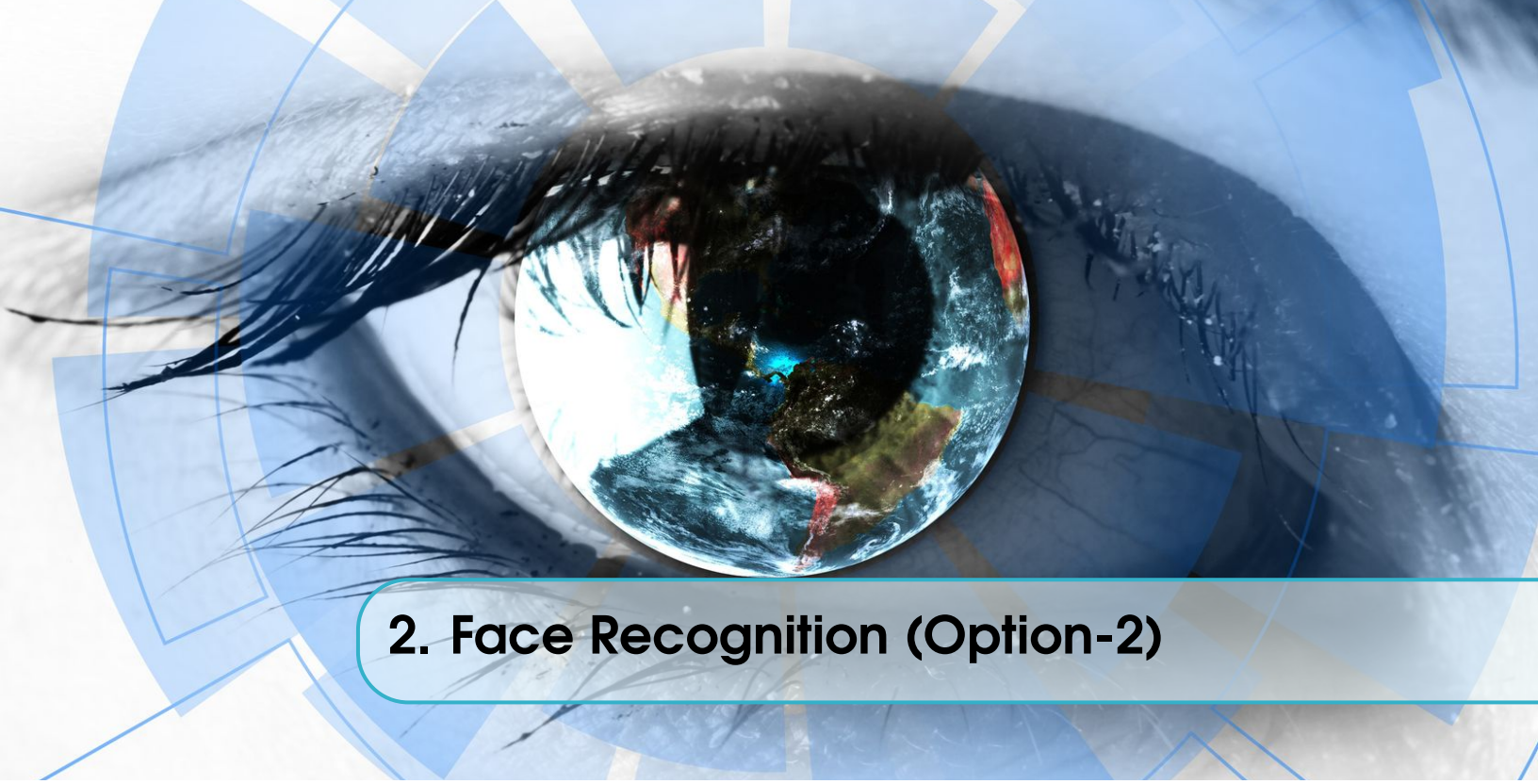
- The output video should be of the same length as the provided video clip.
- The motions of your own character should be determined by the original marionette.
- The usage of libraries is only permitted for I/O purposes and low-level mathematical operations.

1.4 Deliverables

You are asked to create a zip file of all deliverables, and submit it via the Canvas system by 7:00AM, Monday (Week 12), 22 October, 2018. Your assignment will only be marked if all the deliverables can be accessed through the Canvas System. Late submissions will not be accepted.

- All the related source code and a runnable demo program.
- A PDF technical report formatted with LaTeX.
- A live demo in week 12 during lab time and an example output video.

[1] Hints: Consider controlling the movement speed of your objects after interactions.



2. Face Recognition (Option-2)

2.1 General

This is an individual assignment (Option-2) and worth 25% of the total assessment of this unit of study. In this assignment, you are required to build a prototype of a real-time face recognizer. You are required to implement a face recognition framework with Eigenface and multi-scale sliding window. However, you are welcome to try out different alternatives. Some supporting material and data will be provided throughout the semester during lab time. 3% bonus marks will be given to exemplary solutions. You are welcome to use ANY programming languages to complete your assignment.

2.2 Main Objectives

2.2.1 Data Preparation

(Baseline) To train a face classifier, you need to prepare true positive and negative image patches. You can find them from the internet or create your own dataset.

2.2.2 Face Classifier

[10%] Train a face classifier to distinguish faces and non-face patches using the Eigenface method and a k-nearest-neighbour classifier. Some parameters, for example the number of eigenfaces or the number of neighbours, may not only effect the recognition accuracy, but also have an influence on the processing speed. You may need to set up those parameters through experiments. An example of Eigenface is shown in Fig 2.1. However you are welcome to discover more advanced algorithms once you have the required method implemented. Be sure you are able to explain the algorithms clearly in your technical report and explain them clearly to your tutor.

2.2.3 Age Prediction (*updated)

[5%] For each recognised face, build another model to predict the age of him/her and the results can be stored in a text file. In terms of this age prediction, the scale face input can be constant.

Figure 2.1: An example of Eigenface basis.

2.2.4 Multi-Scale Recognition

[5%] Use multi-scale sliding windows to detect a single face in an image using the classifier you build. The detected face should be labelled with a box showing the size and coordinates. You should fine-tune the classification framework according to your face recognition results and consider special cases such as rotation and overlap. Once your framework works for a single person, apply it to an image with many faces.

Figure 2.2: An example face recognition on the image of the Solvey conference 1927.

2.2.5 Real-Time Recognition on Camera Streaming or other Creative Effects

[Bonus] Embed your trained framework with a camera video stream and demonstrate that it works in real-time. An effective demonstration of real-time face recognition is enough and the accuracy is tolerated within a certain range. The real-time recognition is just an example and other creative ideas may be applied here. The marking will depend on your demonstration.

2.3 Technical Report

[5%] Draft a 6-8 page technical report to demonstrate your pipeline. The main sections of report should at least include Introduction, Implementation and Conclusion. The implementation can discuss the algorithm and your experimental results. Your report should be written following the scientific style and formatted with LaTeX. Do not worry if you have not used latex before. It has a similar syntax to HTML. You can find a handy online latex editor at `https://www.sharelatex.com/`. An alternative choice is overleaf which can be found at `https://www.overleaf.com/`.

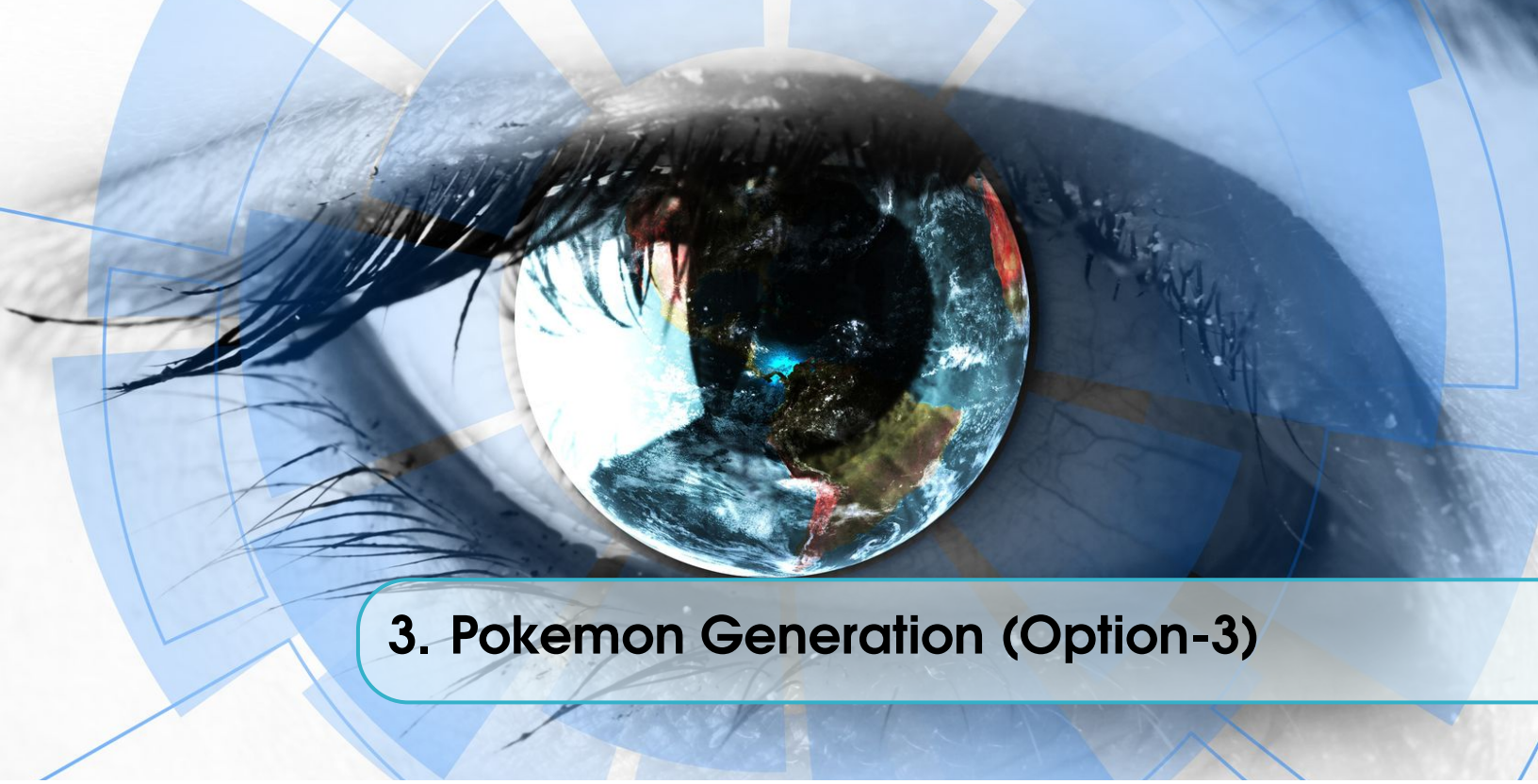
2.4 Constraints

- You are welcome to use any programming languages to complete this assignment. You are encouraged to use low-level mathematical libraries only.
- The usage of other libraries are only permitted if you can demonstrate the algorithms and they are not included in the basic pipeline above.

2.5 Deliverables

You are asked to create a zip file of all deliverables, and submit it via the Canvas system by 7:00AM, Monday (Week 12), 22 October, 2018. Your assignment will only be marked if all the deliverables can be accessed on the Canvas System. Late submissions will not be accepted.

- All the related source code and a runnable demo program.
- A PDF technical report formatted with LaTeX.
- A live demo in week 12 during lab time and a demo video (the demo video should be shorter than 2 minutes and smaller than 100MB).



3. Pokemon Generation (Option-3)

3.1 General

This is an individual assignment (Option-3) and worth 25% of the total assessment of this unit of study. In this assignment, you are required to implement a new Pokemon image generation framework with Wasserstein Generative Adversarial Network (WGAN). The detailed explanation can be found at https://arxiv.org/abs/1701.07875. However, you are welcome to try different alternatives as long as their performance is better than the baseline WGAN. Some supporting materials and data will be provided throughout the semester during lab time. 5% bonus marks will be given to exemplar solutions. You are welcome to use ANY programming languages to complete your assignment.

3.2 Main Objective

3.2.1 Data Preparation

[5%] Neural networks perform well when there are enough training samples. For example, training VGGNet on ImageNet successfully requires more than 140 million images. With given pokemon images, data augmentation techniques can be used to enrich the training dataset for your algorithm. In this part, you need to show your understanding on different data augmentation techniques such as Crop, Pad (top, right, bottom, left), Flip vertically and horizontally.

3.2.2 Discriminator

[5%] Be creative about the design of the discriminator. One of the working examples of a discriminator follows repeated structures of convolutional layer, batch normalization layer and activation layer. After implementing your own discriminator, prove the capability of your discriminator using some small experiments such as a hand-written digits classification task.

3.2.3 Generator

[5%] Be creative with the design of the generator. One of the working examples of a generator first reshapes input images and then follows a series of deconvolution layers. One example of feature map

dimensions after going through deconvolution layers is $8 \times 8 \times 256$, $16 \times 16 \times 128$, $32 \times 32 \times 64$, $64 \times 64 \times 32$ and $128 \times 128 \times 3$. After implementing your own generator, prove the capability of your generator using some small experiments such as an image segmentation task.

3.2.4 New Pokemon Image Generation

[5%] The network design of WGAN is similar to Generative adversarial network (GAN), which can

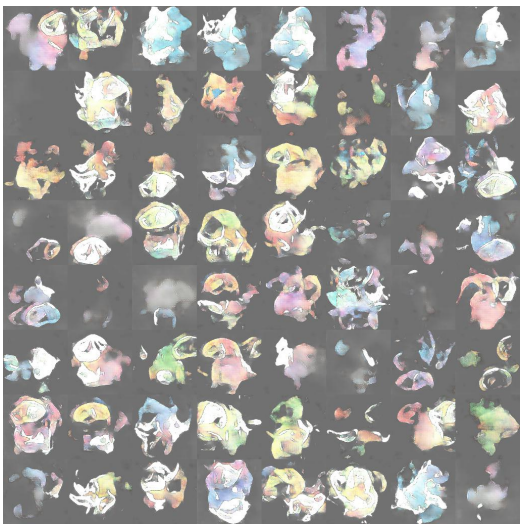


Figure 3.1: An example of new Pokemon images generated by WGAN.

be found in detail at https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.. Demonstrate your understanding of the differences between the cost functions of GAN and WGAN in your report. The cost functions include the discriminator and the generator. Implement the weight clipping and explain why this is necessary in your report. Compare Pokemon image generation results with and without different data augmentation techniques.

3.3 Technical Report

[5%] Draft a high-quality technical report with at least 8 pages. The main sections of the report should at least include Introduction, Method, Experiment, Conclusion and References. The Introduction section should at least discuss the related work, challenges of the given problem and a summary of proposed implementation. The Method section should introduce your implementations in details. Diagrams can help you present your ideas clearly. The Experiment section should discuss results and try to explain the reasons behind them using scientific language. Your report should be written following the scientific style and formatted with LaTeX. Do not worry if you have not used latex before. It has a similar syntax to HTML. You can find a handy online latex editor at `https://www.sharelatex.com/`. An alternative choice is overleaf which can be found at `https://www.overleaf.com/`. The assignment template can be found at `https://eccv2018.org/wp-content/uploads/2018/07/eccv2018camerareadykit.zip`.

3.4 Deliverables

You are asked to create a zip file of all deliverables, and submit it via the Canvas system by 7:00AM, Monday (Week 12), 22 October, 2018. Your assignment will only be marked if all the deliverables can be accessed on the Canvas System. Late submissions will not be accepted.

- All the related source code and a runnable demo program.
- A PDF technical report formatted with LaTeX.
- A live demo in week 12 during lab time and a demo video (the demo video should be shorter than 2 minutes and smaller than 100MB).