GRADUATE CERTIFICATE: Pattern Recognition Systems (PRS)
PRACTICE MODULE: Project Proposal

Date of proposal:

14 Sept 2020

Project Title:

VisionLab

Sponsor/Client: *(Name, Address, Telephone No. and Contact Name)*

Institute of Systems Science (ISS) at 25 Heng Mui Keng Terrace, Singapore
NATIONAL UNIVERSITY OF SINGAPORE (NUS)
Contact: Dr. ZHU FANGMING
Email: isszfm@nus.edu.sg

Background/Aims/Objectives:

Artificial intelligence, as the current direction of technological development and innovation, has given rise to many emerging technologies. For instance, the face recognition technology, which is now widely used in various fields. It is getting closer to our daily lives, such as mobile phones unlock, community security upgrades and missing person search and rescue etc. However, at the same time, it has also triggered to the public’s unease and concerns. Because there are still many uncertainties and risks associated with the application of facial recognition technology.

Out of interest, we would like to design an intelligent web system including facial images generator and detector. The generator can produce colorful virtual/ fake human and cartoon faces, while detector is able to detect the authenticity of human faces that users uploading.

The main purpose of generator is to provide many types of face materials for customers to use as will and no need to worry about face copyright issues. We can image that areas of legitimate use of facial photos are starting to emerge. Artwork and marketing departments can save large budget on expensive photo shoots, instead get enough faces of underrepresented groups to balance media and marketing materials. In addition, more diverse cartoon faces have important artistic and practical values since it can be precious pictures for creation. For example, it helps artists to portray the ultimate fantasy for people to see. General speaking, requesting professionals to draw exclusive cartoon avatars for everyone is impractical and limits creativity.

On the other hand, the detector can help people to distinguish dubious use of applications that have brought attention to AI-generated images: profile pictures for fake news and dating services. And it is also suitable for airport, customs and other ID verification scenarios when using face detection for identity verification.

While upload a clear facial photo to our system, users are able to verify its authenticity and generate corresponding cartoon face. In the meantime, getting lots of various AI-generated human faces for commercial or personal use.

Note,
The definition of "virtual/ fake" is that facial photos are AI-generated.
The definition of "real" is that facial photos are produced by camera.

Requirements Overview:

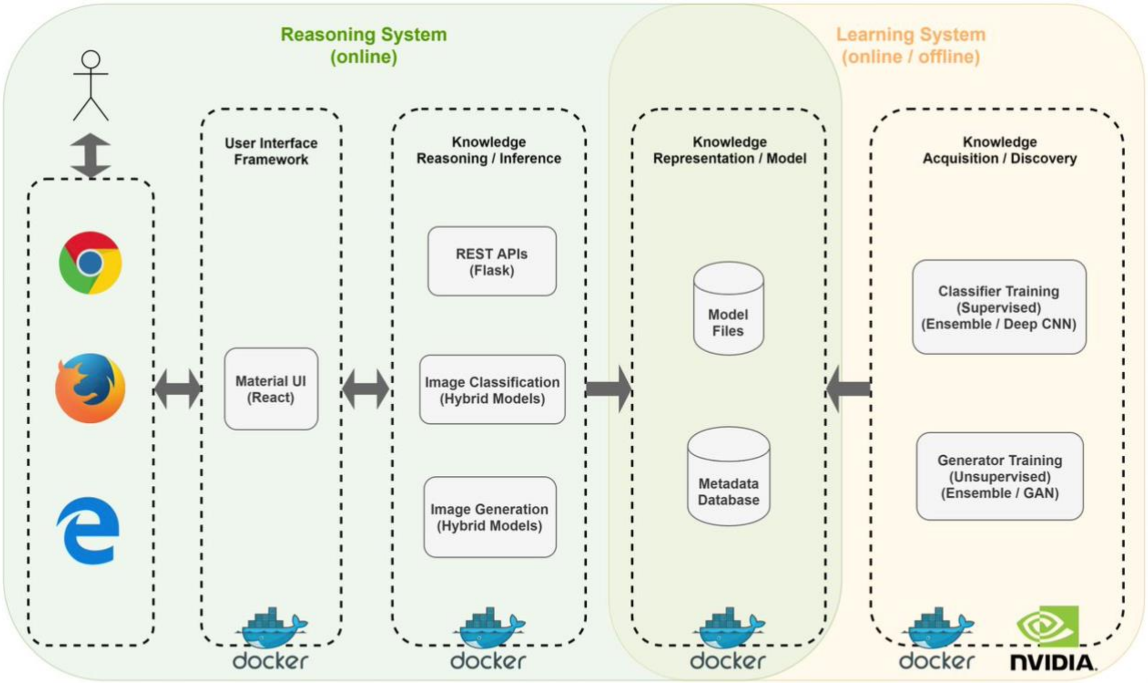
- **Project Objectives**
  - To build an image processing toolbox with the following functionalities:
    - Generate virtual/ fake human and cartoon face.
    - Detect facial photos to determine the authenticity of human faces.
    - User can interact with the system via click several buttons on web UI.

- **Project Scope**
  - Human & cartoon facial image generator
    - Unsupervised Learning
    - Generative Adversarial Network (e.g. DCGAN)
    - Ensemble methods with several models
  - Real/ Virtual human facial photo detector
    - Supervised Learning
    - Deep Convolutional Neuron Network (e.g. ResNet)
    - Ensemble methods with several models
  - Model Training Pipeline
    - Deep Learning pipeline can automatically train by itself and update generative/ predictive models whenever new data uploading.
  - Web UI & REST API
    - Expose Deep Learning models via both Web UI and REST API.

- **Data Sources (tentative)**
  - The dataset can come from ImageNet, Kaggle and GitHub.
  - To collect 10000+ colorful human/ cartoon facial photos, the ratio of real and virtual photos is 1:1.
  - Human face images from the real wild world produced by a real camera with a real person as a subject in the age span of 0-116 years.
  - All facial images are unobstructed and clear without such phenomenon of motion blur (e.g. trailing shake), reflection and shadow.
  - The face in the photo is vertical, and angles of profile, pitch and deflection are less than 30°, 15° and 15° respectively.
  - The picture size of all photos can be 64x64/ 128x128/ 256x256.
  - The picture format is .jpg, .jpeg, .png.

- **System Design**

  The overall system architecture is as below.


Reasoning System (online)
Learning System (online / offline)
User Interface Framework
Material UI (React)
Knowledge Reasoning / Inference
REST APIs (Flask)
Image Classification (Hybrid Models)
Image Generation (Hybrid Models)
Knowledge Representation / Model
Model Files
Metadata Database
Knowledge Acquisition / Discovery
Classifier Training (Supervised) (Ensemble / Deep CNN)
Generator Training (Unsupervised) (Ensemble / GAN)
docker
docker
docker
docker
NVIDIA

Resource Requirements (please list Hardware, Software and any other resources)

➢ **Data Sources (facial photos):**

1) Real human faces
   - *Resolution: 64x64/ 128x128/ 256x256*
   - *No. of images: 5000 - 10000*
2) Virtual human faces
   - *Resolution: 64x64/ 128x128/ 256x256*
   - *No. of images: 5000 - 10000*
3) Cartoon faces
   - *Resolution: 64x64/ 128x128/ 256x256*
   - *No. of images: 5000 - 10000*
4) Realtime photos that user uploads
   - *Resolution: 64x64/ 128x128/ 256x256*
   - *Type of images: Any (balanced for real/virtual)*

➢ **Hardware proposed for consideration**

- Any local/cloud system able to host Docker container.
- Decent GPU to accelerate training process.

➢ **Software / Framework proposed for consideration**

1) Front-end
   - *Web Brower (Chrome/IE/Firefox)*
   - *Nginx (Web Server)*
   - *React (Material UI)*
2) Reasoning Back-end
   - *Flask (REST API)*
   - *Tensorflow (version 2.0+)*
   - *Scikit-learn*
3) Learning Back-end

- *TensorFlow-GPU (version 2.0+)*
- *Scikit-learn*

- **Overall application runtime container**
  - Docker (with GPU support)

- **Programming language**
  - Javascript, HTML, CSS
  - Python3

Number of Learner Interns required: (Please specify their tasks if possible)

A team of three project members:

- Yin Tian Shi - Team lead, overall system design and modelling, technical architecture design and implementation.

- Yang Lu Yi - Team member, overall system design and modelling, use case and algorithm design and implementation.

- Yu Yu - Team member, overall system implementation and validation, project management and documentation.

Methods and Standards:

- **Business Use Case Design**

Use cases for image generator and detector are as below:

No.	Use Case	Functionality	User Action	System Output
1	Virtual Human Facial Image Generator	End users are able to generate 10 number of random fake human faces each time. If needed, can generate more photos when click “generate more” button.	Users select AI-generated faces from drop list on Generator page.	A gallery view of human faces will show up on web UI, and user can download photos selectively or by batches.
2	Virtual Cartoon Facial Image Generator	End users are able to generate 10 number of random fake cartoon faces each time. If needed, can generate more photos when click “generate more” button.	Users select AI-generated faces from drop list on Generator page.	A gallery view of cartoon faces will show up on web UI, and user can download photos selectively or by batches.
3	Real/ Virtual Facial Photo Detector	End users upload an photo via web UI, and a robot will suggest whether the photo is AI-generated.	Users upload photos via a drop-down box on detector page.	Classification result will print on screen, possibly with explanations.
4	Hybrid Models & User Feedback	Web UI displays corresponding images that are generated by different models (e.g. model A/ B/ C), and those models can be selected by users with checkboxes.	Users select no. of models from checkboxes.	The following info will show up on web UI. 1) *Types of models that generate images*. 2) *Users’ preference of different models*.

				3) *The history of image downloads from each model.*

➢ Data Sources Preparation

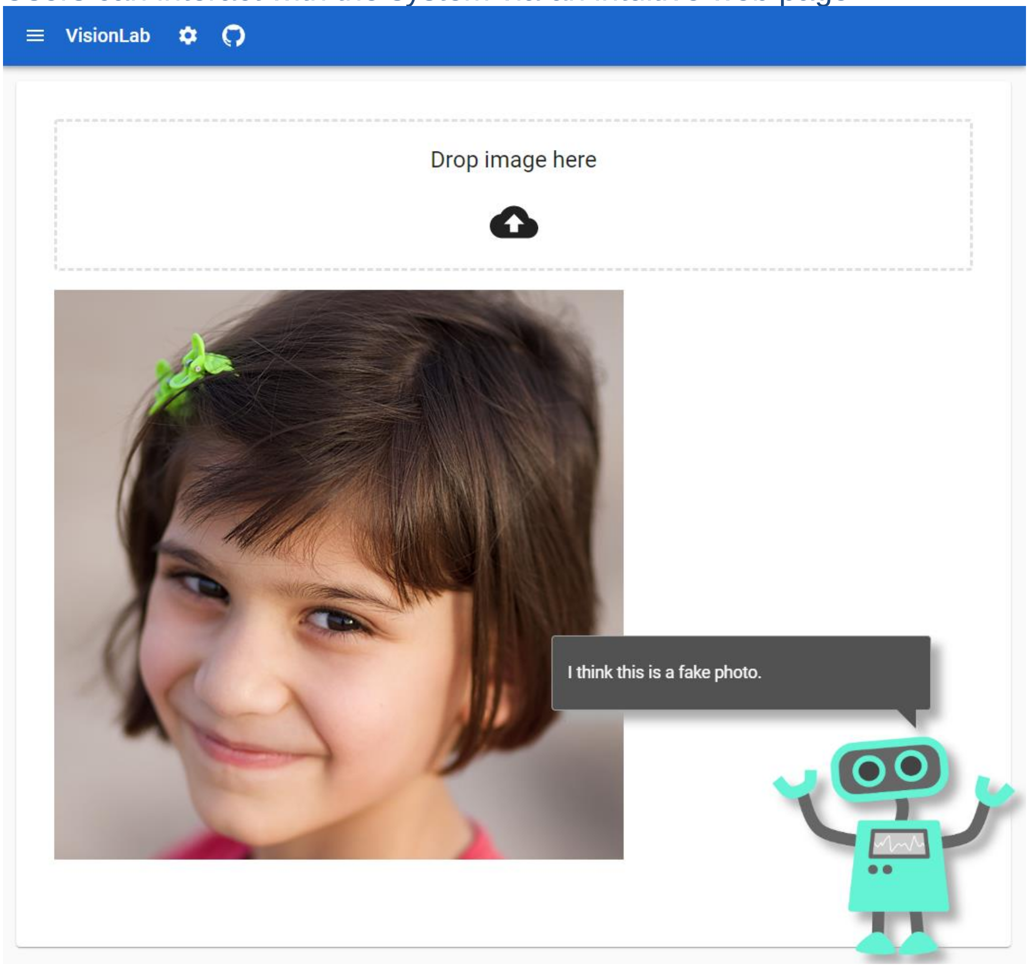
The specific datasets are to be updated as model training is still under implementation.

➢ Model Training and Evaluation

1) Run model training against prepared image datasets.
2) Gather statistics/ metrics for the training process and trained model.
3) Evaluate and adjust non-trainable hyperparameters of the model.
4) Repeat training (from a refresh model) until all hyperparameters are tuned to achieve acceptable model performance.
5) With fixed non-trainable hyperparameters, run model training for a few times and save the model with the best performance.
6) Repeat above steps for other model definitions (of different neural network topology)

➢ System UI Design (mock-up)

Users can interact with the system via an intuitive web page.



I think this is a fake photo.

➢ System Backend Structure

Real/fake image detection:

1) Photo uploaded is transferred to reasoning back-end via REST API.
2) Grab models (trained and saved by learning back-end) from shared data repository, and re-instantiate them.
3) Run prediction against each of the model instances and get an average prediction score.
4) Formulate the detection result (with explanations) based on prediction score and return results to front-end.

- **System Iteration/ Improvement**

After the system goes live, it will continue to collect valuable data that users uploading. Since those are most significant and useful data for application scenarios.

1) Re-run the training process periodically with both pre-collected dataset photos and user uploaded photos. Compare the results and replace models when there is improvement.
2) Search and identify other possible model definitions (e.g. from newly published papers). Try them out and add suitable models to the collection of saved models.

Team Name: Project Group 2
Project Title (repeated): VisionLab
System Name (if decided): VisionLab
Team Member 1 Name: Yu Yu
Team Member 1 Matriculation Number:
Team Member 1 Contact (Mobile/Email):
Team Member 2 Name: Yin Tian Shi
Team Member 2 Matriculation Number:
Team Member 2 Contact (Mobile/Email):
Team Member 3 Name: Yang Lu Yi
Team Member 3 Matriculation Number:
Team Member 3 Contact (Mobile/Email):

For ISS Use Only

Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned:		