MASTER OF TECHNOLOGY
(INTELLIGENT SYSTEMS)

INSTALLATION & USER GUIDE

VisionLab
(Facial Images Generator & Detector)

GROUP MEMBERS

YIN TIAN SHI

YANG LU YI

YU YU

Contents

1.0 System Overview

The VisionLab is an intelligent web system that including facial images generator and facial images detector. The generator can produce colorful virtual/ fake human (e.g. Asian face, celebrity and online celebrity faces etc.) and cartoon faces, while detector is able to detect the authenticity of facial images that users uploading or produced by generator. Both generator and detector integrate various models, users are able to interact with the system while changing settings (no. of generated images per batch, model types) to choose favourite models. Furthermore, you also can mark and download any photo on generator, in the meantime, there is a cute robot responds to you whether that uploaded photo is real or AI-generated.

2.0 System Installation

2.1 Recommended Browsers

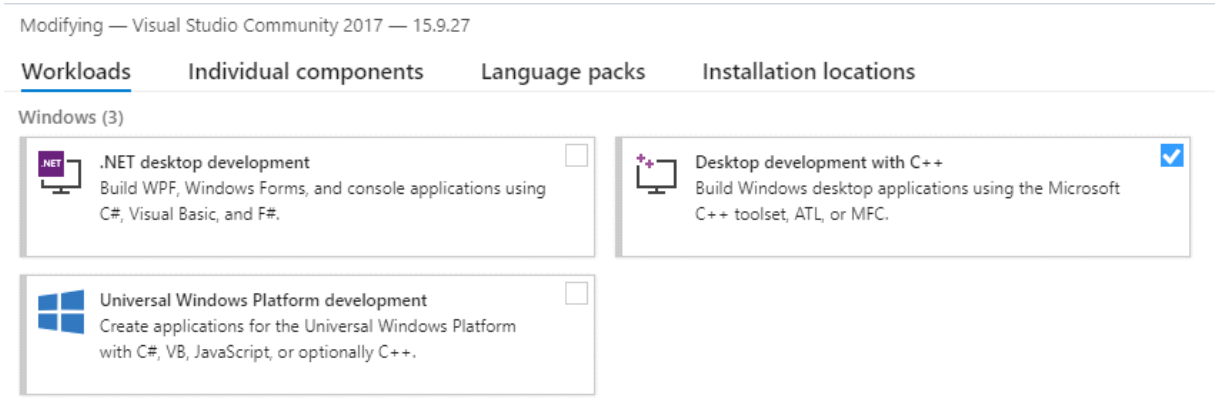
The system web UI supports the following Web browsers:

- Google Chrome Version 59 and above
- Microsoft Edge 44 and above
- Firefox 75 and above
- Safari Version 10 and above

2.2 Environment Requirement

The system deploys to any environment having Docker Engine installed. Optionally, public internet connection is recommended, in order to support all functionalities in the system. Both Linux and Windows platform are supported. Linux is recommended for performance and compatibility reasons.

1. **Installation on Linux (recommended for production)**
   - Docker Engine https://docs.docker.com/engine/install/
   - NVIDIA Container Toolkit https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html#docker
   - Please note that NVIDIA Container Toolkit is required for running StyleGAN models and training pipelines online.
2. **Installation on Windows**
   - Docker https://docs.docker.com/get-docker/
     Note, if you have already installed Oracle VM VirtualBox on laptop, please download Docker Desktop/ Toolbox https://docs.docker.com/toolbox/toolbox_install_windows/
   - Python 3.6 above installation. Recommend to create an environment on Anaconda3.
   - One or more high-end NVIDIA GPUs (at least 16GB DRAM), latest NVIDIA drivers, CUDA 10.0 toolkit and cuDNN 7.6.5. Access to this URL for detailed introduction. https://www.tensorflow.org/install/gpu
   - Desktop development with C++ via Visual Studio Community 2017 installer https://visualstudio.microsoft.com/vs/older-downloads/



- Required Python Packages are as follows.
  pip install tensorflow-gpu==1.14.0
  pip install scipy==1.3.3 requests==2.22.0 Pillow==6.2.1
  pip install flask==1.1.2 flask_cors==3.0.9
  pip install opencv-python-headless==4.4.0.44
  pip install mysqlclient==2.0.1 SQLAlchemy==1.3.20

2.3 Deployment

The system images are pulled from Docker Hub registry. As our system integrates various models to provide more diverse face images and flexible operation for users, majority of trained models can be triggered successfully once docker and docker-compose working on your laptop. In particular, StyleGAN models and training pipelines will be disabled if no NVIDIA support in Docker.

Note, detailed user guide is written based on Windows platform.

➢ **Deployment on Linux**

1) Install Docker Engine & NVIDIA Container Toolkit.
2) Access to our GitHub https://github.com/IRS-3Y/Vision-Lab/tree/master/SystemCode and download file "vlab-compose.yml" to a local directory.
3) Go to the local directory and execute the command on a terminal to start the application.
   ```
   E:\Vision-Lab-master\SystemCode>docker-compose -p vlab -f vlab-compose.yml up -d
   ```
   Note, it will take a while to pull all necessary data in the first time.
4) VisionLab application is now running and available on http://localhost
5) Execute the command on a terminal to stop the application.
   ```
   E:\Vision-Lab-master\SystemCode>docker-compose -p vlab -f vlab-compose.yml down -v
   ```

➢ **Deployment on Windows**

1. Docker & Docker-Compose

1) Verify Docker and Docker-compose have installed successfully.
   ```
   (base) C:\Users\User>docker -v
   Docker version 19.03.8, build afacb8b

   (base) C:\Users\User>docker-compose -v
   docker-compose version 1.25.4, build 8d51620a
   ```

2) Check what images include in Docker.

```
(base) C:\Users\User>docker images
```

Before you pull images, it shows nothing.

```
C:\Users\15229>docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
```

After you pull images, it shows all available images.

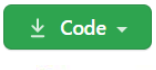
```
REPOSITORY                      TAG                 IMAGE ID            CREATED             SIZE
irs3y/vlab-backend-dcgan        latest              6f78186f6785        27 hours ago        4.26GB
irs3y/vlab-backend-stylegan     latest              f28c1c25ae8e        27 hours ago        5.22GB
irs3y/vlab-backend-core         latest              0b01669ae2c7        27 hours ago        3.82GB
irs3y/vlab-frontend             latest              42db9e36bfa3        39 hours ago        122MB
mysql                           5.7.26              e9c354083de7        15 months ago       373MB
```

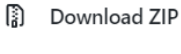
2. Environment setup for StyleGAN models implementation

1) Install Desktop development with C++ via Visual Studio Community 2017.
2) Install latest NVIDIA drivers, CUDA 10.0 toolkit and cuDNN 7.6.5.
   Note, if you have installed more than one CUDA & cuDNN version, please only save path of CUDA 10.0 and cuDNN 7.6.5 on system environment variables.
3) Create an environment on Annaconda3

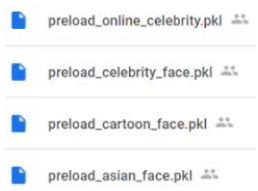
- Open Anaconda Prompt and execute the command to set up environment.
  ```
  (base) C:\Users\User>conda create --name tsflow114 python=3.6
  ```
  Note, tsflow114 & python version can be revised.
- Execute command to activate the environment.
  ```
  (base) C:\Users\User>conda activate tsflow114
  ```
  Note, the environment has changed to (tsflow114)
- Check python version with command
  ```
  (tsflow114) C:\Users\User>python --version
  Python 3.6.12 :: Anaconda, Inc.
  ```
- Install required Python Packages.
  pip install tensorflow-gpu==1.14.0
  pip install scipy==1.3.3 requests==2.22.0 Pillow==6.2.1
  pip install flask==1.1.2 flask_cors==3.0.9
  pip install opencv-python-headless==4.4.0.44
  pip install mysqlclient==2.0.1 SQLAlchemy==1.3.20
  ```
  (tsflow114) C:\Users\User>pip install tensorflow-gpu==1.14.0
  ```

3. Pull Docker Images and Start Application

1) Access to our GitHub https://github.com/IRS-3Y/Vision-Lab and click 

-> 

you will see that a zip file named " Vision-Lab-master" is downloading.



2) Unzip the file to a local directory, and access this link to download 4 files.
https://drive.google.com/drive/folders/1HQAJHXJhJZ2DdEFMUob3_L9uCISj8pCM?usp=sharing

Then put them into this directory (*e.g. E:\Vision-Lab master\SystemCode\backend\instance\models\stylegan2*).

3) Go to the local directory *(e.g. E:\Vision-Lab-master\SystemCode\backend)* and execute the command on a terminal.

```
(tsflow114) E:\Vision-Lab-master\SystemCode\backend>set FLASK_APP=backend && flask run --port=5003
```

When you see the following command, please go to step 4).

```
  np_resource = np.dtype([("resource", np.ubyte, 1)])
 * Running on http://127.0.0.1:5003/ (Press CTRL+C to quit)
```

4) Go to the local directory that put the Docker Compose configuration (vlab-compose-windows.yml) and execute the command on a terminal.

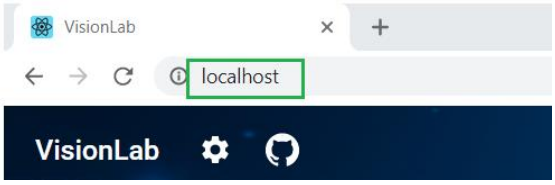
```
E:\Vision-Lab-master\SystemCode>docker-compose -p vlab -f vlab-compose-windows.yml up -d
```

Note, it will take a while to pull all necessary data.

```
Digest: sha256:fc8d072a05259e394dc5c174c6e044312b1cc9a00469032419502 40f015c935a
Status: Downloaded newer image for irs3y/vlab-frontend:latest
Creating vlab_database_1 ... done
Creating vlab_backend_stylegan_1 ... done
Creating vlab_backend_dcgan_1    ... done
Creating vlab_backend_1          ... done
Creating vlab_frontend_1         ... done

(tsflow114) E:\Vision-Lab-master\SystemCode>
```

5) Our VisionLab application is now running and available on http://localhost



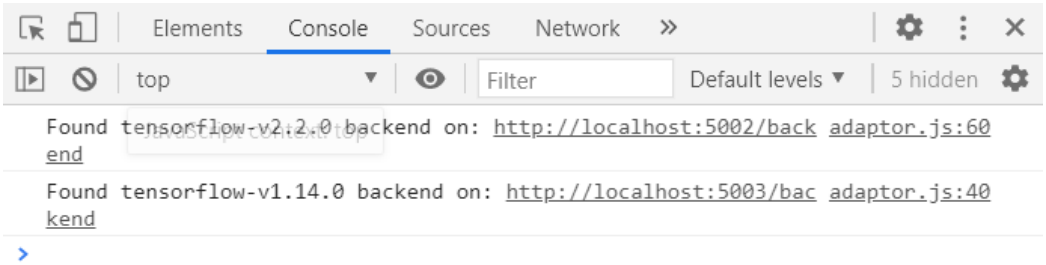
Note, if you installed Docker Toolbox, then the VisionLab may not be running on http://localhost, in that case, please execute the command on a terminal to get IP:

docker-machine ip

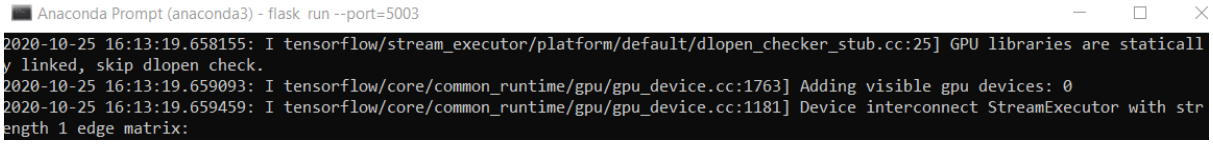
Then the application will show up on http://{DOCKER_HOST_IP}

```
D:\My Lab\EKGS>docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER     ERRORS
default   *        virtualbox   Running   tcp://192.168.99.100:2376           v19.03.5
```

6) Right click on the VisionLab web, select "Inspect" option and click "Console" button, you will see two different version of Tensorflow backends are working.



In the meantime, you also can monitor on-going operations on web system via that terminal you executed command `docker-compose -p vlab -f vlab-compose-windows.yml up -d`



4. Trouble-shoot Connection Problems

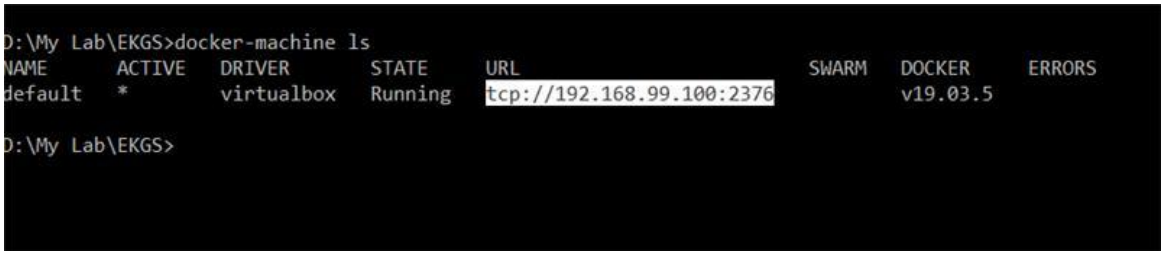
1) Network ports occupied

For application to startup successfully, it requires port 80 is not pre-occupied by other system processes. In case this port is occupied, you may change the port mapping by modifying vlab-compose.yml file in text editor (before start application).

To change web UI port, update the mapping for port 80
e.g. "80:80" => "8080:80", web UI will then be accessed via http://localhost:8080

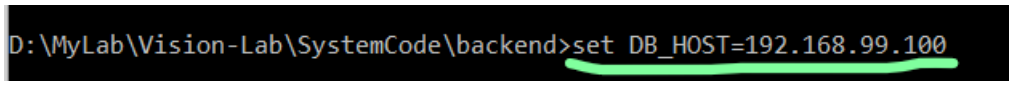
2) Docker is not running on localhost

If your docker is not running on localhost, and you need to get IP of docker-machine by the following command



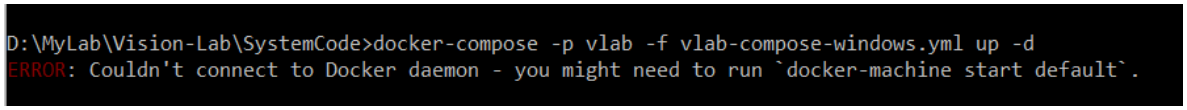
Then you need to set DB_HOST=the IP address of your docker machine by typing below command before you start Windows backend service for StyleGAN2

set DB_HOST=192.168.99.100

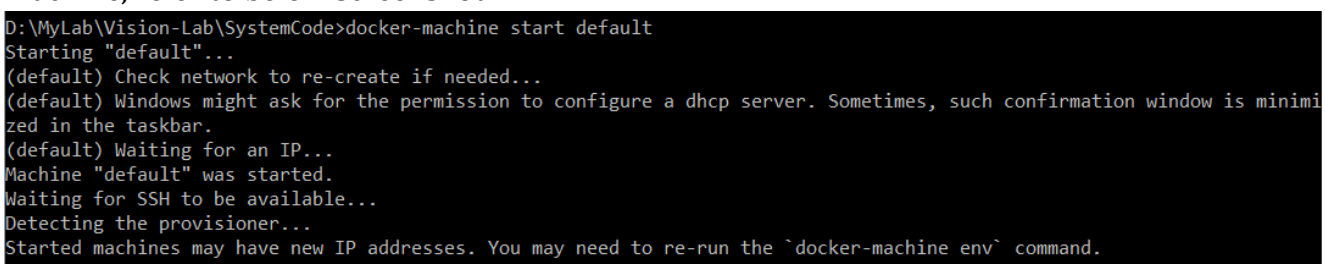


3) Couldn't connect to Docker daemon

You may encounter below error when you first start up the application using this command "docker-compose -p vlab -f vlab-compose-windows.yml up -d".



In this case, please execute the command "docker-machine start default" to start up the docker machine, refer to below screenshot:



5. **Stop Application**

Execute the command on a terminal to stop the application.

```
E:\Vision-Lab-master\SystemCode>docker-compose -p vlab -f vlab-compose-windows.yml down -v
```

```
(tsflow114) E:\Vision-Lab-master\SystemCode>docker-compose -p vlab -f vlab-compose.yml down -v
Stopping vlab_frontend_1          ... done
Stopping vlab_backend_1           ... done
Stopping vlab_backend_dcgan_1     ... done
Stopping vlab_backend_stylegan_1  ... done
Stopping vlab_database_1          ... done
Removing vlab_frontend_1          ... done
Removing vlab_backend_1           ... done
Removing vlab_backend_dcgan_1     ... done
Removing vlab_backend_stylegan_1  ... done
Removing vlab_database_1          ... done
Removing network vlab_default
Removing volume vlab_entities

(tsflow114) E:\Vision-Lab-master\SystemCode>
```

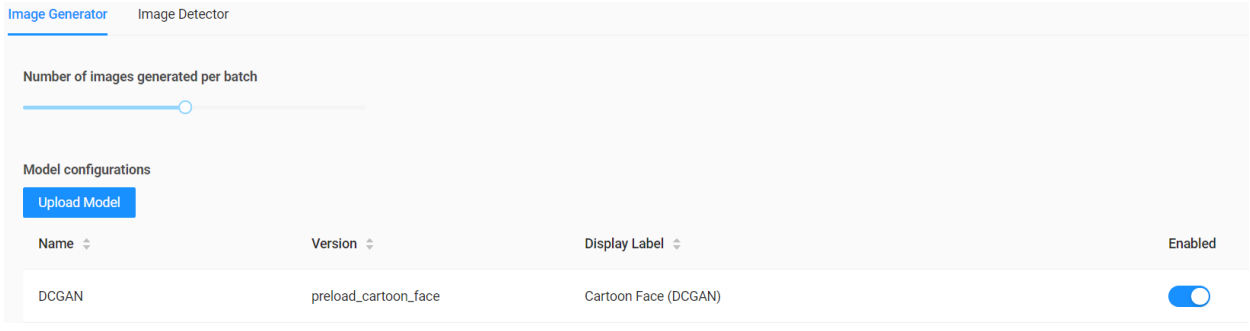
3.0 System Settings & User Guide

Open up your preferred browser and go to the URL http://localhost or "http://192.168.x.x" as shown below.

3.1 System Settings

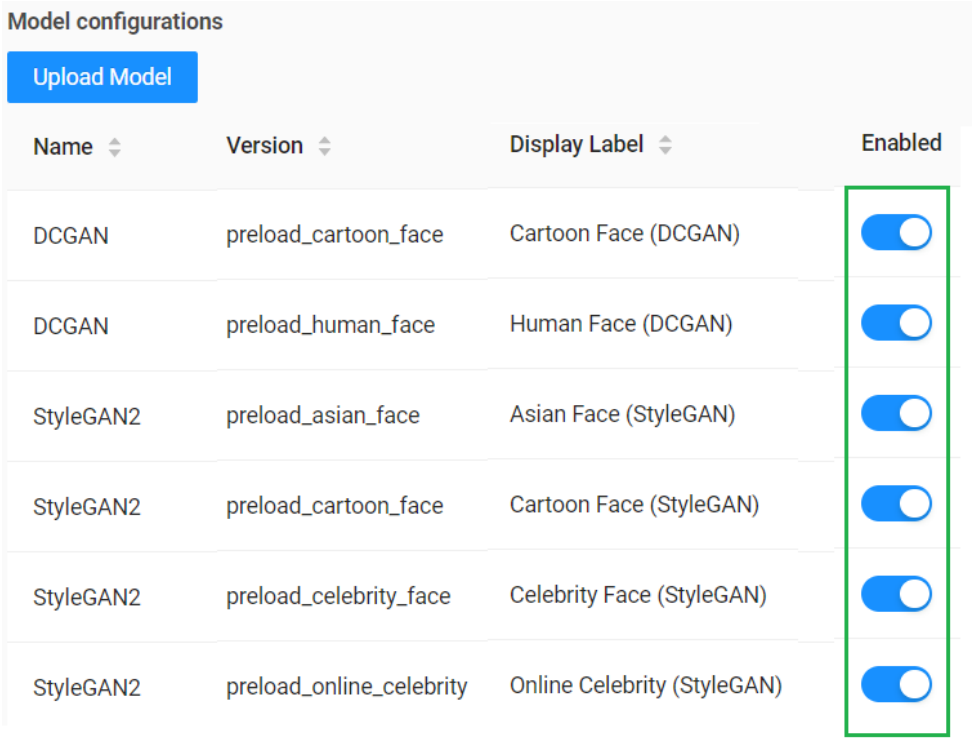
1) Click this setting icon on top right of web.

2) The first default setting is for Image Generator.

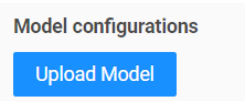


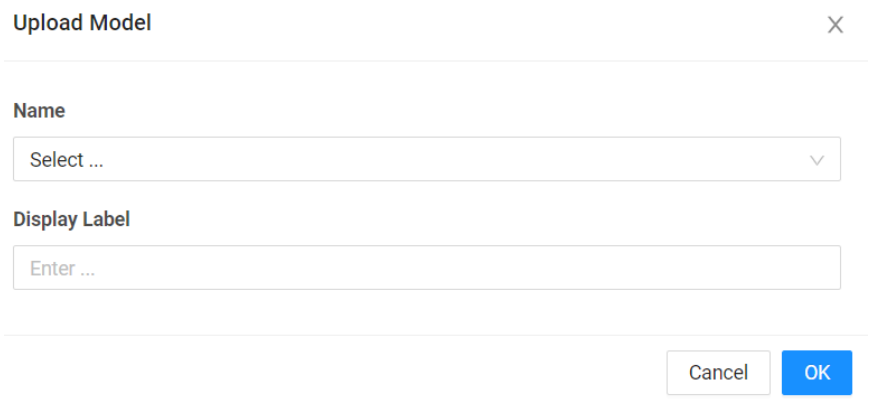
- On the setting page of image generator, while moving the bule circle to left/ right, you can define the quantity of generated images per batch. The range of value is 0 (minimum) - 20 (maximum). The no. Set by default is 10.

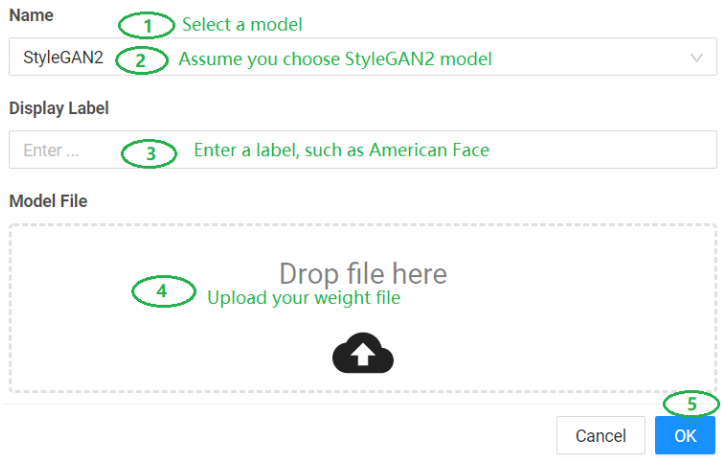
- On the 'Model configuration' part, you can see all available models are enabled by default. You can disable any one or more models below. But I do not recommend you to do it. Because if you disable one model, you will not see it on Image Generator page. Furthermore, there is also enable/ disable option of each model on Image Generator page.



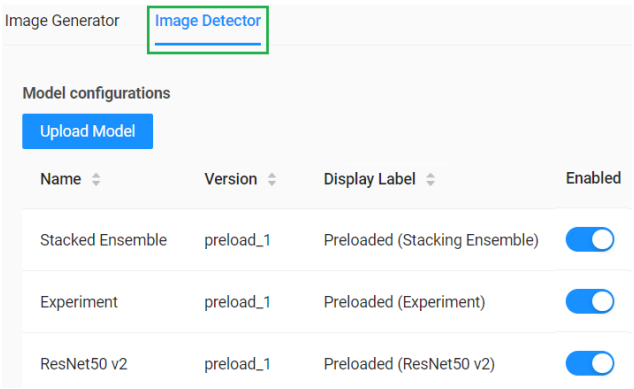
- When you click "Upload Model" button on 'Model configuration' part, you will see a pop-up page 'Upload Model'. If you train another dataset (e.g. American face) by using our DCGAN or StyleGAN model, you are allowed to upload your weight file here. Please follow the step to proceed it.

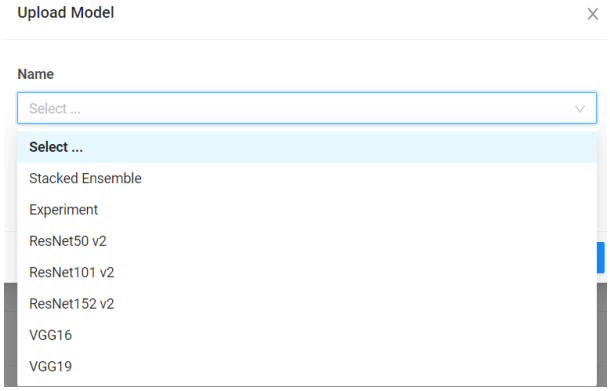






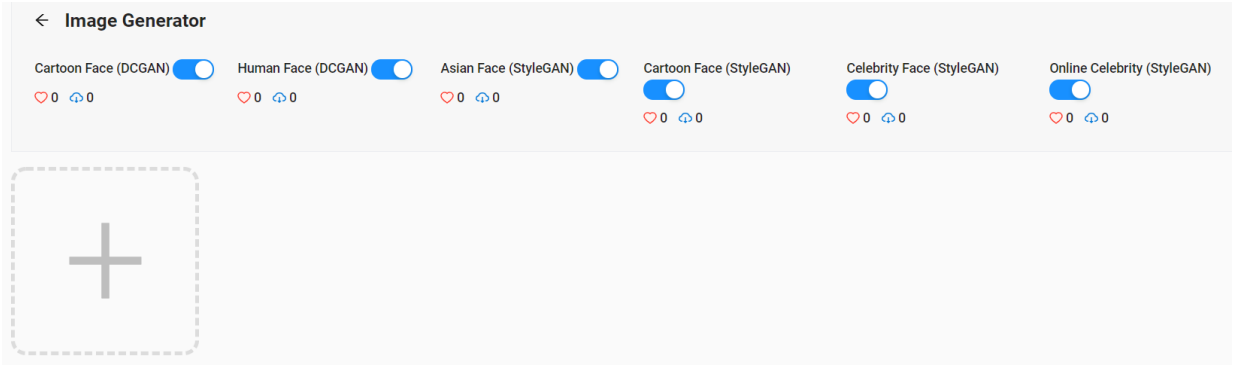
3) Click “Image Detector” option, you will go to its setting page. Please note that we only enabled several better models for generator. If you click “Upload Model” button, you will see more available models for you to train your dataset. Similarly, please refer to above steps to upload your weight file for generator.



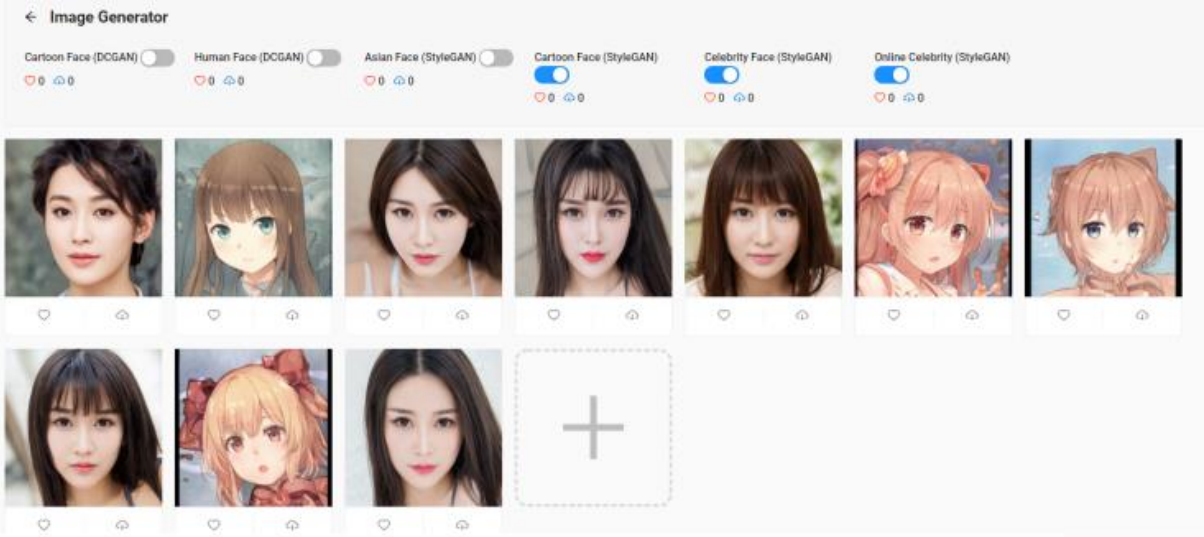


3.2 Image Generator

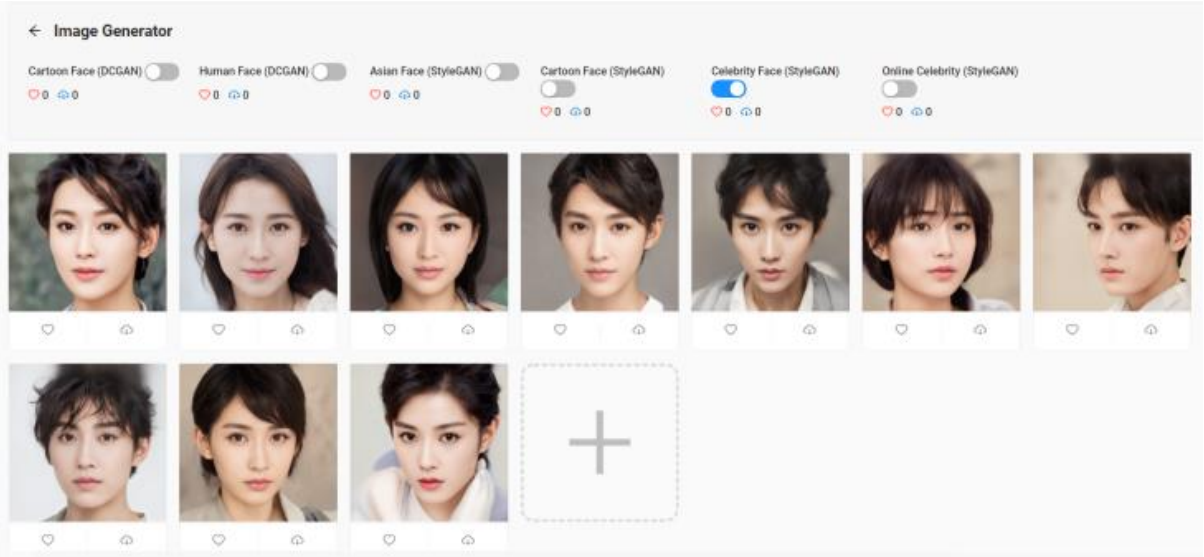
1) Click “System settings” to go back home page, and click “Image Generator” icon, you will see the whole page as below.



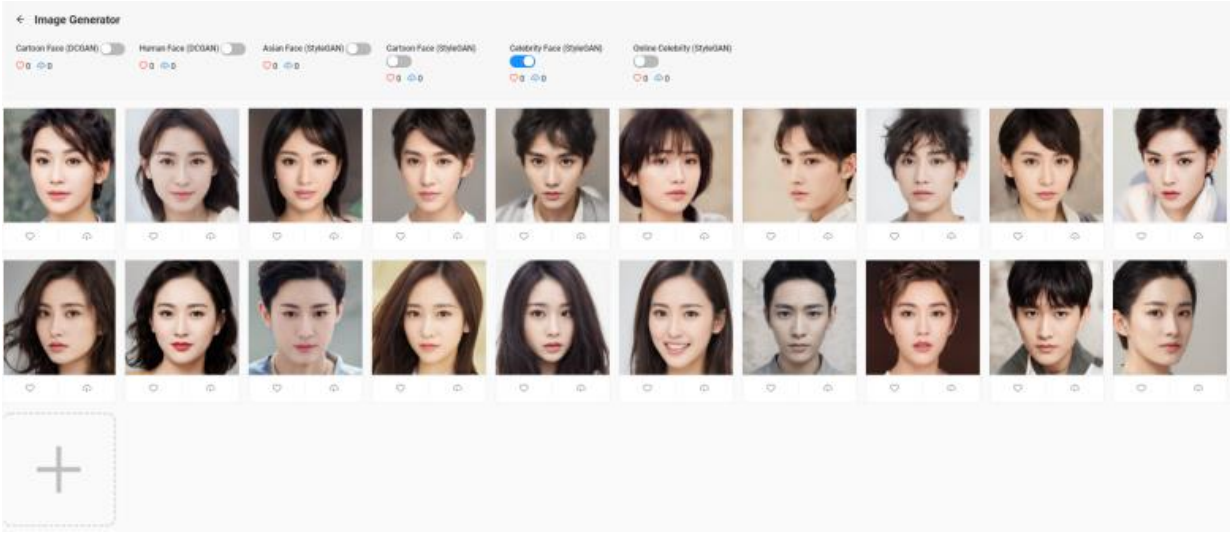
2) Now all models are enabled by default, you can disable any one or more of them, then click + to generate fake faces. For instance, I disabled the first three models and click +, then mixed cartoon faces, celebrity faces and online celebrity faces are generated, total is 10 as I used the default no.



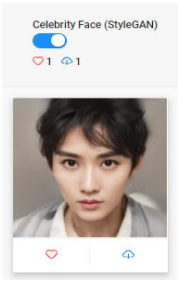
3) In addition, I can enable one model only, such as celebrity face (StyleGAN).



4) Click + icon can generate another 10 images. There is no same face no matter how many images you generated.



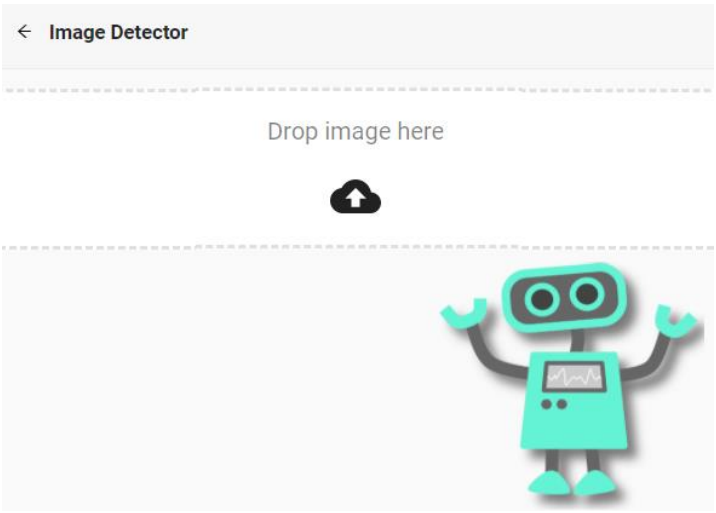
5) If you prefer one face image and want to download it, you can click ♡ and ☁ button at the bottom of that photo. Once you clicked, the no. of same button under Celebrity Face (StyleGAN) increased to 1, 1 respectively. This function is quite helpful for us to investigate the preference of users on those fake faces.



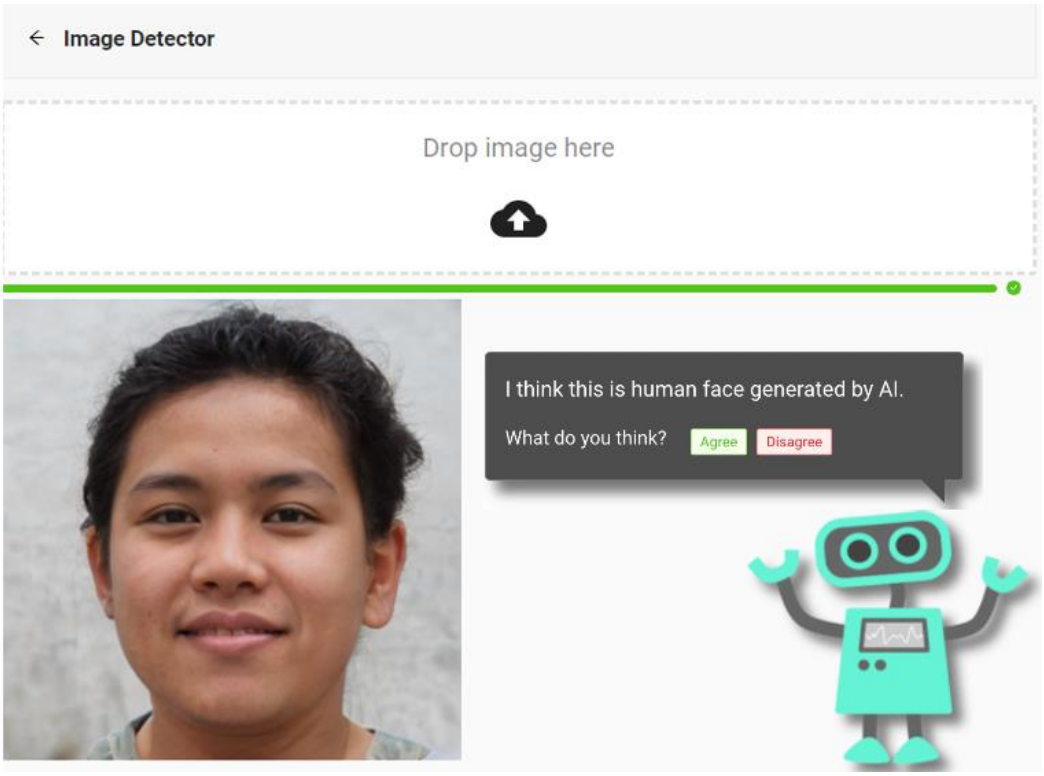
3.3 Image Detector

Click return button on Image Generator page, and click Image Detector icon on home page. The Image Detector page will pop up as below.

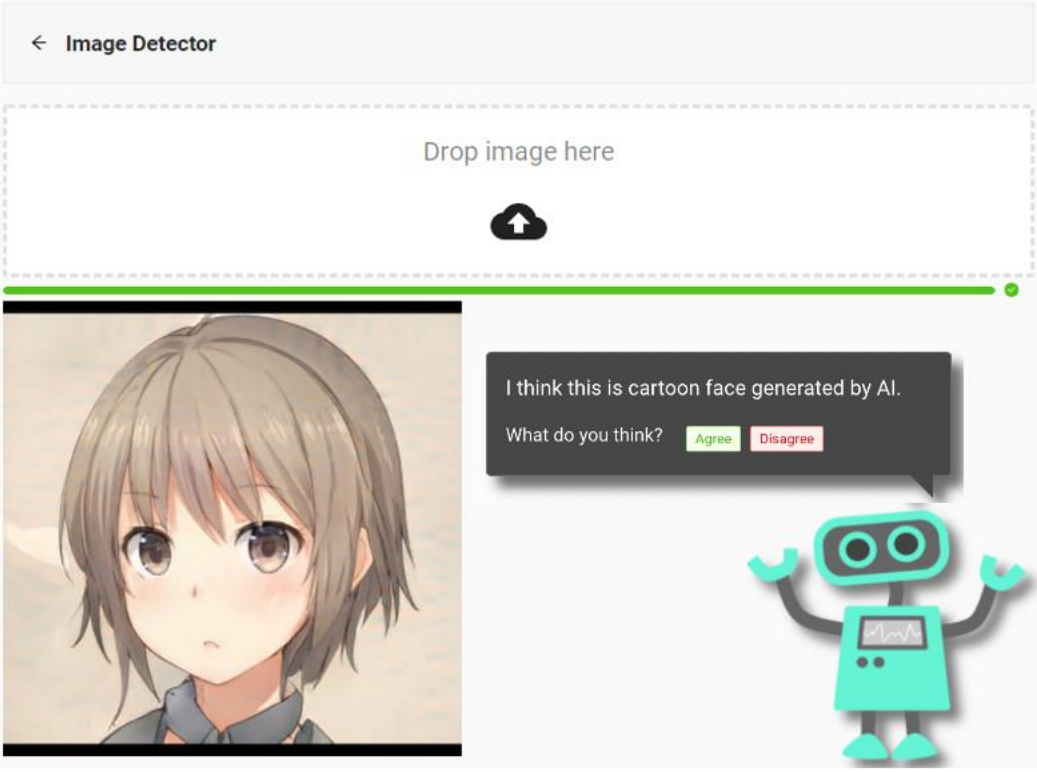
1) Click "Drop image here" and upload a photo to identify it is real face or AI-generated. There are some examples that I tried.



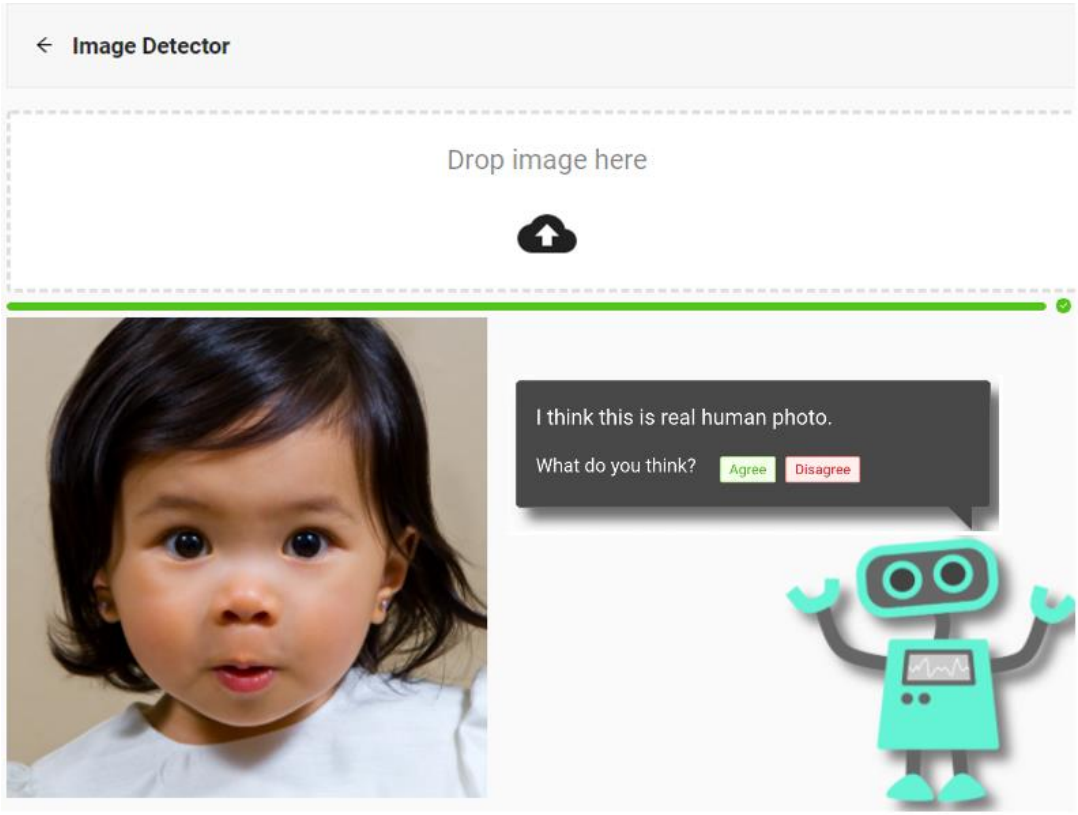
- When I uploaded a fake human face that generated from our Image Generator, the robot will say "I think this is human face generated by AI. What do you think?"



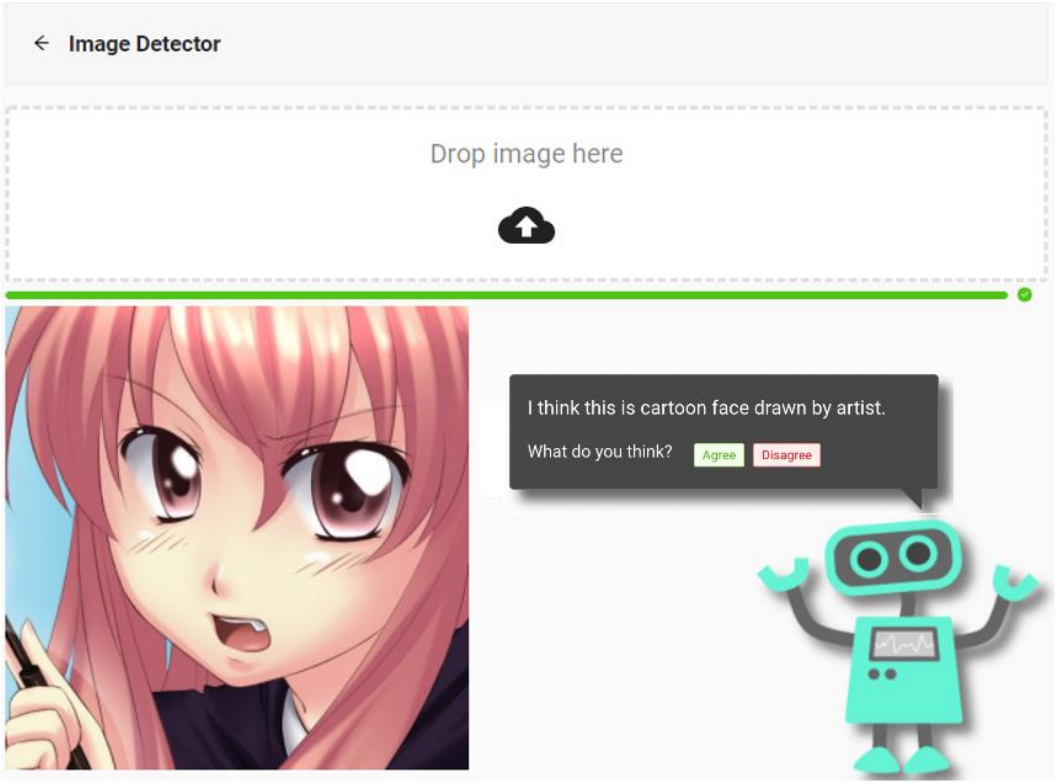
- When I uploaded a fake cartoon face that generated from our Image Generator, the robot will say “I think this is cartoon face generated by AI. What do you think?”



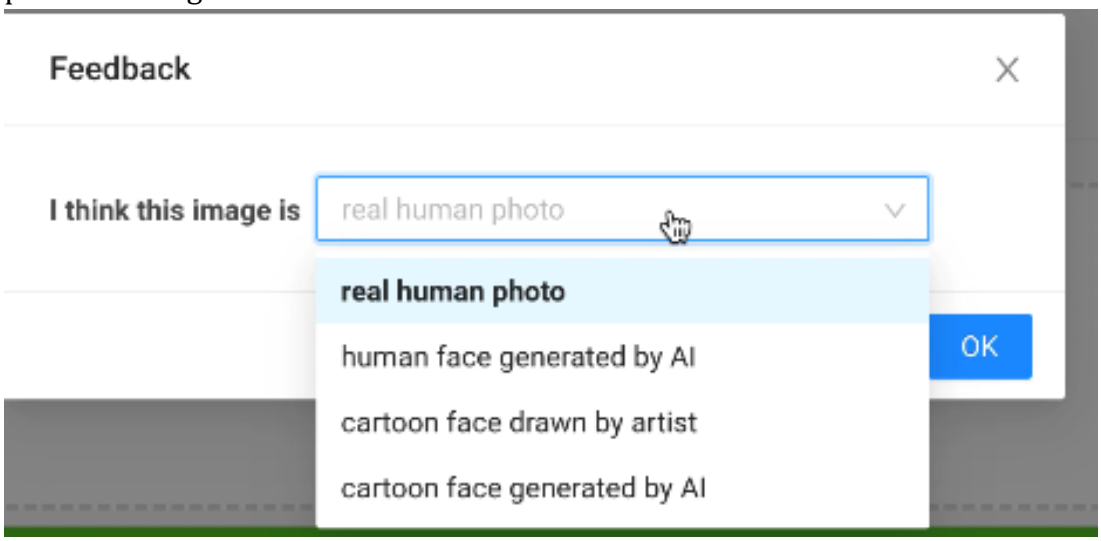
- When I uploaded a real human face that took by a camera, the robot will say “I think this is real human photo. What do you think?”



- When I uploaded a real cartoon face that drawn by artist, the robot will say “I think this is cartoon face drawn by artist. What do you think?”

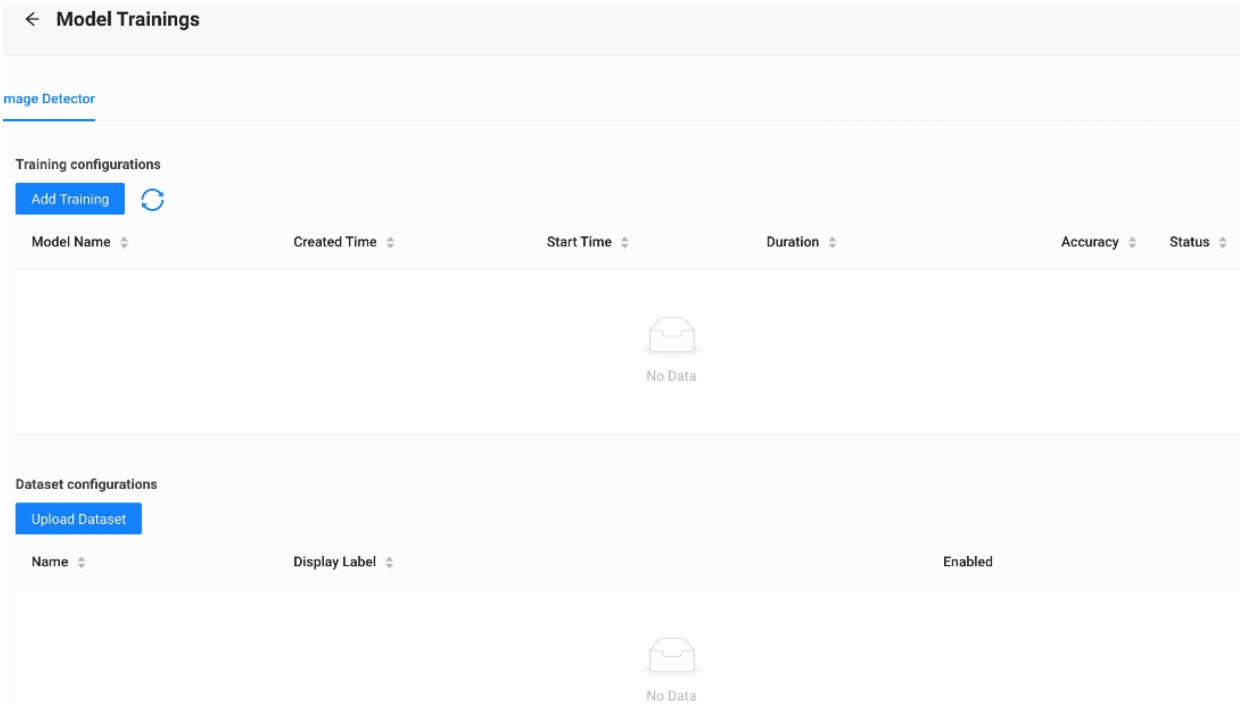


2) When you click “Agree” button, that picture will be auto added into corresponding dataset. When you click “Disagree” button, you can help feedback to us which dataset should that picture belong to.

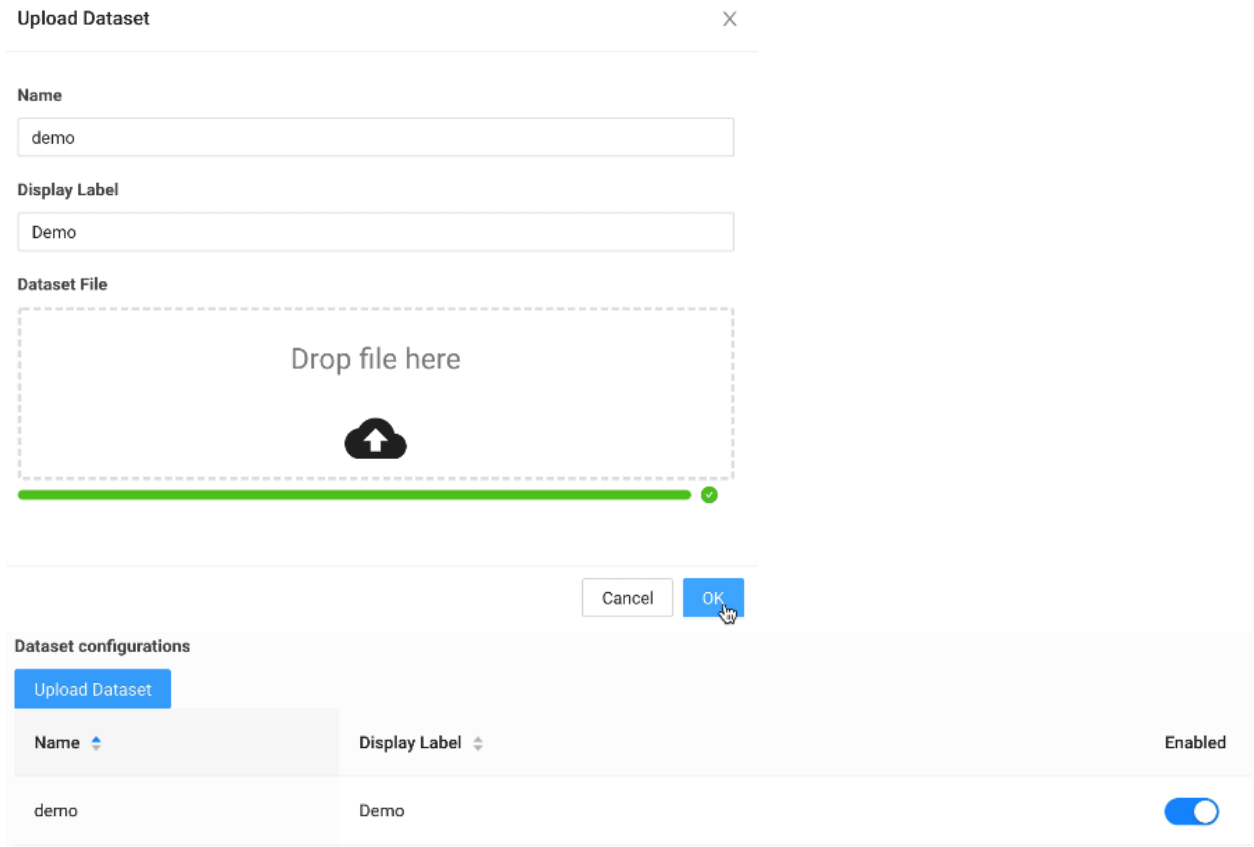


3.4 Model Trainings

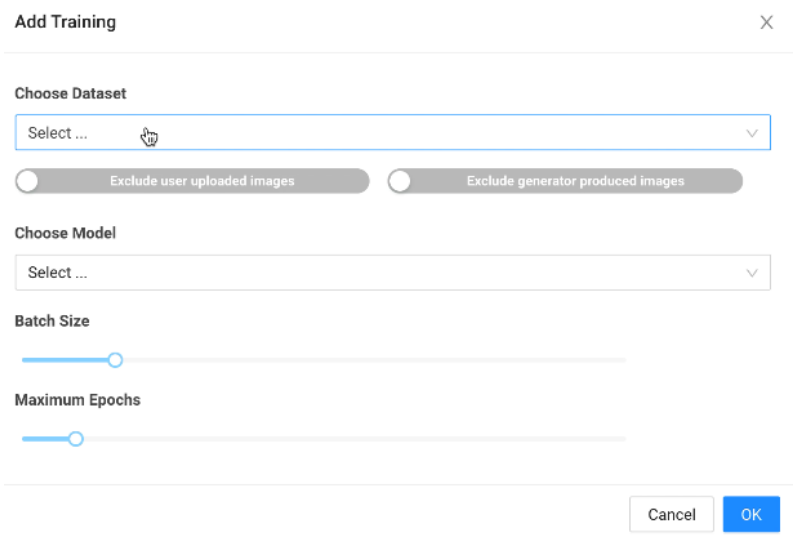
1) Click the following small icon on top left corner of web system. The Model Trainings page will pop up.



2) There are two parts, one is training configuration, the other one is dataset configuration. Firstly, click "Upload Dataset" button to upload training dataset, and fill up Name and Display Label info.



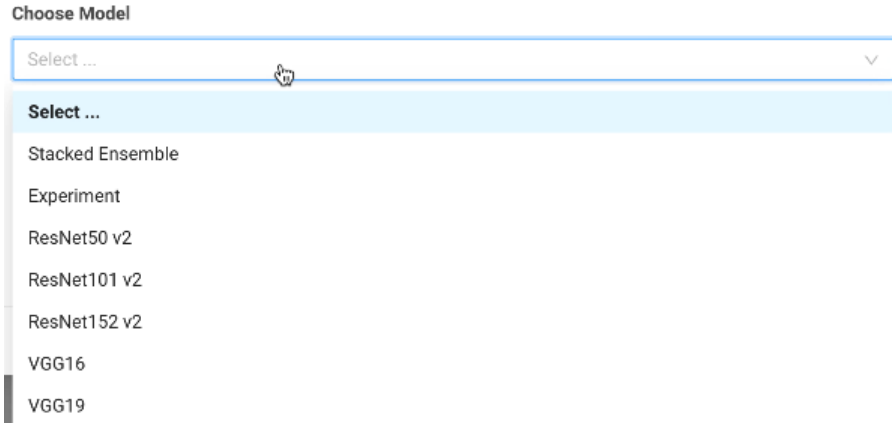
3) Secondly, click "Add Training" button and then complete necessary steps on this page before starting training.

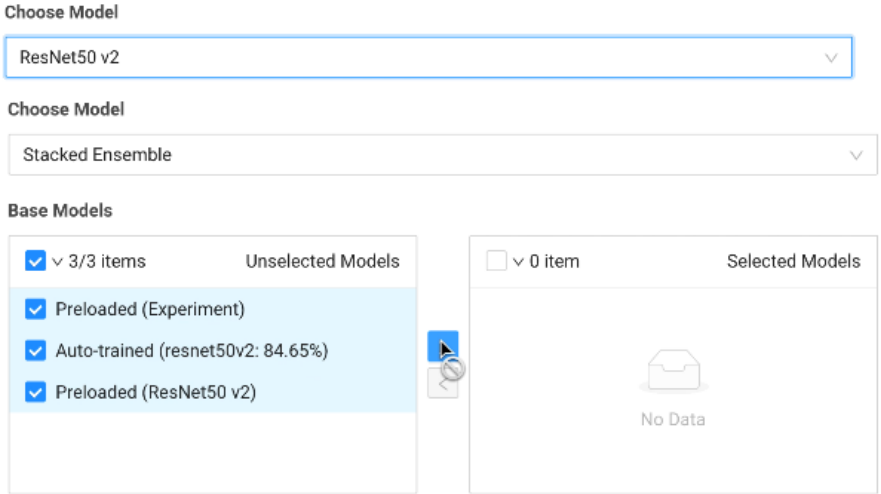


- Select uploaded dataset

- Decide to include/ exclude user uploaded images & generator produced images.

- Choose Model (We chose *ResNet50 v2* & *Stacked Ensemble* to illustrate.)



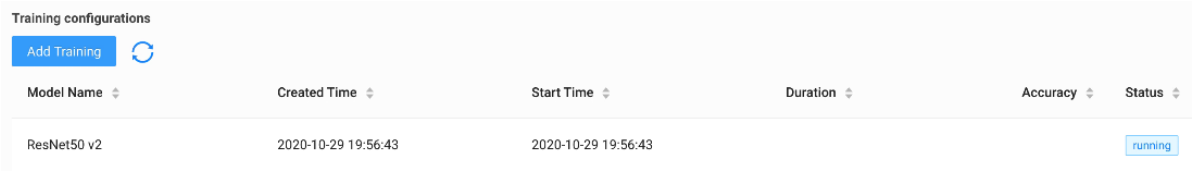


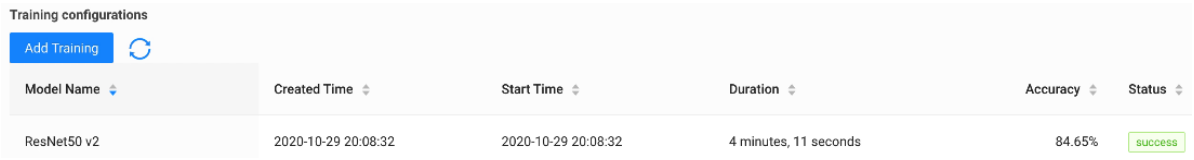
- Set Batch Size

- Set Maximum Epochs

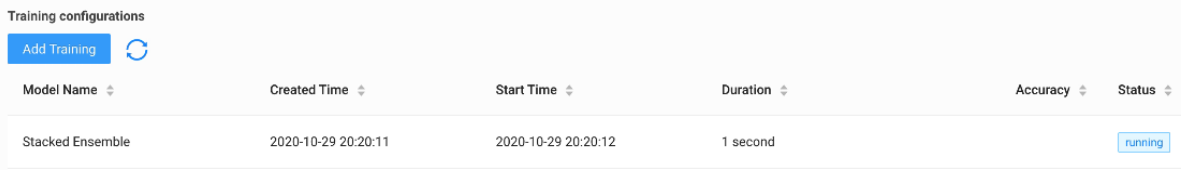
4) After completed the above settings, then you will see the model is training.

- *ResNet50 v2*

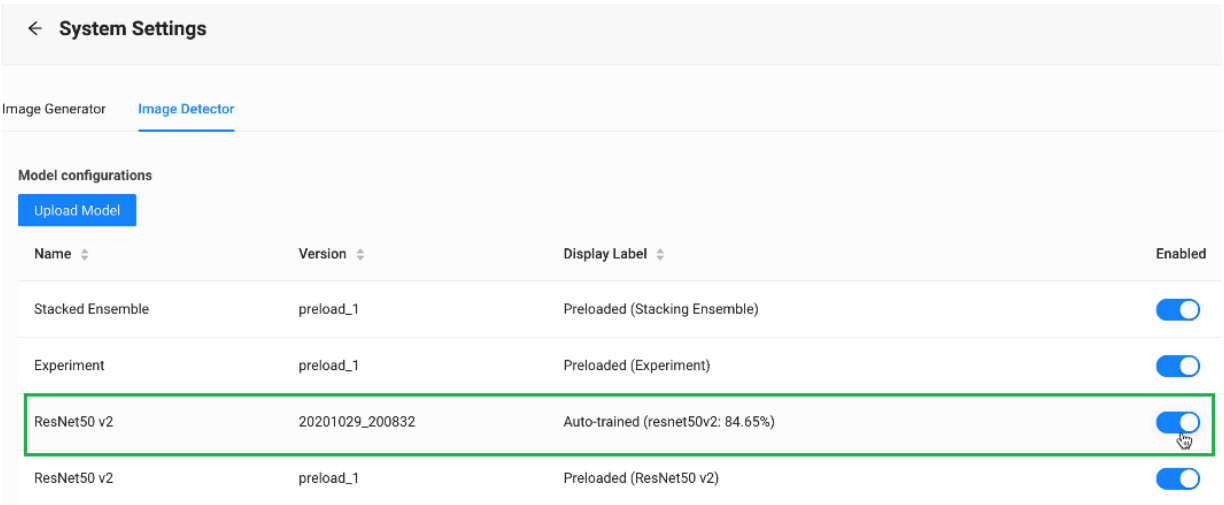




- *Stacked Ensemble*

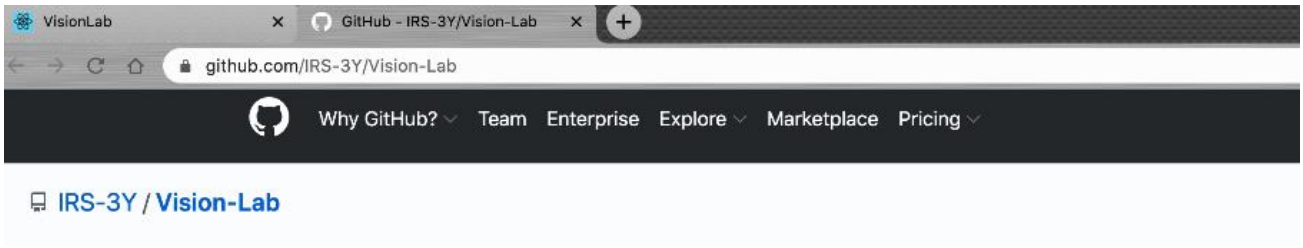


5) Once done, go to "System Settings", the just trained model has added to Image Detector. As it is disabled by default. If you want to use it, please enable it.



3.5 Project Site

If you want to learn more about our system VisionLab, please access to our GitHub link via clicking the following project Site icon.


VisionLab
GitHub - IRS-3Y/Vision-Lab
github.com/IRS-3Y/Vision-Lab
Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
IRS-3Y / Vision-Lab

Thank you!