

Workshop Series on Python Frameworks

Image Retrieval with Web Interface using Flask

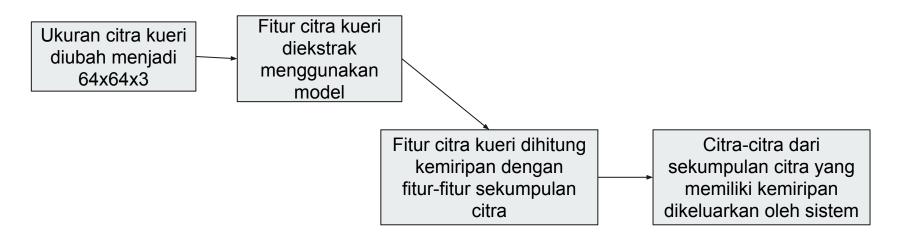


Sekilas tentang *Image Retrieval*

- Sebuah sistem untuk memperoleh citra atau gambar dari sekumpulan citra berdasarkan *query* yang berupa teks atau bisa sebuah citra.
- Permasalahan yang biasa terjadi adalah ketidaksesuaian hasil yang diharapkan dengan hasil yang dikeluarkan oleh sistem.
- Pada workshop ini, input dan output yang digunakan adalah citra.

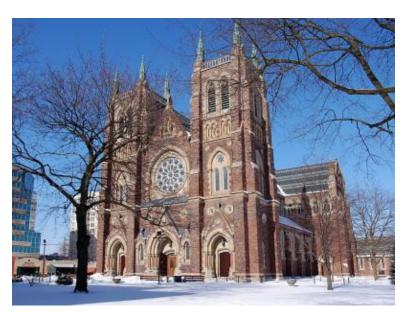


Aliran Program





Contoh Fitur Hasil Ekstraksi



Citra Input (64 x 64 x 3)

10	8	3	4	5	10	7	9
8	5	5	6	3	1	9	3
5	4	4	5	4	9	1	8
3	8	10	2	7	10	10	5
5	5	0	3	2	6	9	9
9	5	4	4	7	8	7	7
9	9	6	4	7	6	3	8
2	8	6	5	7	9	1	2

Fitur Hasil Ekstraksi (8 x 8 x 1)

Demo file di getModelCNN.ipynb



Sekilah tentang Ekstraksi Fitur

- Pada sistem Image Retrieval yang akan dibuat menggunakan ekstraksi fitur untuk mengetahui fitur penting dari citra kueri dan sekumpulan citra-citra.
- Pada proses ekstraksi fitur, sistem yang dibuat menggunakan model CNN (Convolutional Neural Network) sederhana yang sudah dilatih atau training menggunakan library tensorflow.
- Pengukuran kemiripan menggunakan Euclidean Distance. https://en.wikipedia.org/wiki/Euclidean_distance



Lapisan CNN Sederhana

Model: "model_1"

Layer (type)	Output Shape	Param #	
input_1 (InputLayer)	[(None, 64, 64, 3)]	0	
conv2d (Conv2D)	(None, 64, 64, 32)	896	
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0	
conv2d_1 (Conv2D)	(None, 16, 16, 64)	18496	
max_pooling2d_1 (MaxPooling2	(None, 8, 8, 64)	0	
conv2d_2 (Conv2D)	(None, 8, 8, 1)	577	

Total params: 19,969 Trainable params: 19,969 Non-trainable params: 0



Sekilas tentang Flask

- Flask merupakan *framework* aplikasi web ringan yang ditulis dengan Python. Flask dapat digunakan sebagai web server.
- Di dalam projek Flask terdapat beberapa file dan folder penting:
 - File_utama.py
 - Templates
 - Static
 - o dll.
- Dokumentasi lengkap https://flask.palletsprojects.com/en/1.1.x/.



Instalasi Flask (Windows)

• Buka command prompt atau power shell dan masukan perintah:

```
C:\Users> pip install flask
```

• Untuk mengetahui keberhasilan instalasi masukan perintah:

```
C:\Users> python
>> from flask import Flask
```

• Jika tidak ada informasi error, maka berhasil.



Membuat Projek Sederhana

- Buat sebuah folder dan beri nama folder tersebut, misal app-image-retrieval.
- Buat file Python dan beri nama file tersebut app.py
- Mari kita mulai untuk melakukan coding, yay 😊 👼



Menjalankan Web Server Pertama Kali (1)

• Buka app.py, tambahkan kode untuk mengimport pustaka Flask, Tensorflow, Numpy dan Pillow yang dibutuhkan pada workshop ini.

```
from flask import Flask, render_templat, request
from tensorflow.keras.models import load_model
import numpy
```

- Jika pustaka selain Flask belum terinstal, silakan install terlebih dahulu.
 - Tensorflow : https://www.tensorflow.org/install/pip
 - Numpy : https://pypi.org/project/numpy/
 - Pillow : https://pypi.org/project/Pillow/



Menjalankan Web Server Pertama Kali (2)

Tambahkan kode untuk mendeklarasikan pustaka flask.

```
app = Flask(__name__)
```

• Tambahkan kode untuk membuat url/route dan fungsi untuk memberikan aksi yang terjadi jika url tersebut diakses.

```
@app.route('/')
def index():
    Return "Hello guys!"
```



Menjalankan Web Server Pertama Kali (3)

• Tambahkan kode sebuah kondisi untuk mengetahui bahwa file tersebut merupakan file utama yang akan didalam beserta kode yang ada di dalamnya.

```
If __name__ == __main__:
    app.run(debug=True)
```

Kemudian simpan file.



Menjalankan Web Server Pertama Kali (4)

Buka Command Prompt, kemudian masukan perintah:

```
nama_direktori> python app.py
```

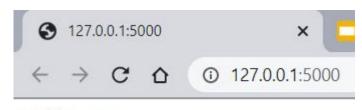
• Muncul informasi mengenai alamat server yang sedang berjalan untuk app yang kita buat.

```
* Serving Flask app "app" (lazy loading)
* Environment: production
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 934-816-635
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



Menjalankan Web Server Pertama Kali (5)

• Setelah itu, buka aplikasi browser dan masukan alamat url 127.0.0.1



Hello Guys!



Program *Image Retrieval* Sederhana(1)

- Buka kembali file app.py
- Tambah kode dua variabel global yang menyimpan path dari dataset dan menyimpan nya dalam bentuk list/array

```
folder_dataset = "./static/assets/images/dataset/"
LIST_PATH_IMAGE_DATASET = [
    f for f in os.listdir(folder_dataset) if os.path.isfile(os.path.join(folder_dataset, f))
]
```

• Tambahkan kode parameter / argumen methods pada fungsi route index yang telah dibuat dengan sebuah nilai berbentuk *list Python* berisi http method yaitu GET dan POST

```
@app.route('/', methods=['GET', 'POST'])
```



Program *Image Retrieval* Sederhana(2)

• Tambahkan kode untuk pengkondisian jika request yang dikirim adalah POST.

```
def index():
   if request.method =='POST':
```

 Tambahkan kode sebuah variabel untuk menyimpan file akan diupload di dalam pengkondisian jika request yang dikirim adalah POST.

```
if request.method =='POST':
    file = requests.file['file']
```



Program Image Retrieval Sederhana(3)

 Tambahkan kode untuk menyimpan citra dalam sebuah variabel dengan pustaka Pillow dan menyimpan citra pada direktori kita

```
if request.method == 'POST':
    ...
    img = Image.open(file.stream)
    upload_img_path = 'static/assets/images/upload' + file.filename
    img.save(upload_img_path)
    ...
```



Program *Image Retrieval* Sederhana(4)

• Tambahkan kode untuk mengubah ukuran citra menjadi 64x64x3, menormalisasi nilai citra tersebut ke dalam skala 0 s.d. 1 dan mengubah ke dalam tipe data *float*-32.

```
if request.method =='POST':
    ...
    img_input = img.resize((64, 64), Image.ANTIALIAS)
    image_input = np.array(img_input).astype('float32') / 255
    ...
```

 Tambahkan kode sebuah variabel yang menyimpan indeks urutan pada array/list pada sekumpulan citra-citra

```
if request.method =='POST':
    ...
    indices_val_of_sim = mainImageRetrieval(image_input=image_input)
    ...
```



Program Image Retrieval Sederhana(5)

• Tambahkan kode variabel yang menyimpan *array/list path* berdasarkan indeks pada variabel yang telah dibuat.



Program Image Retrieval Sederhana(6)

• Tambahkan kode dua variabel yang menyimpan nama file dari citra kueri dan *list path* nama sekumpulan citra-citra.

```
if request.method =='POST':
    ...
    query_img = file.filename
    result = indices_val_of_sim_path
    ...
```

• Tambahkan kode untuk merender template html yang akan dibuat dengan tambahan argumen nama file citra kueri dan kumpulan nama file dari sekumpulan citra atau dataset.

```
if request.method =='POST':
    ...
    return render_template('index.html', query_img=query_img, data=result)
    ...
```



Program *Image Retrieval* Sederhana(7)

• Tambahkan kode menampilkan file html saja tanpa argumen tambahan jika request yang dikirim adalah GET.

```
def index():
    if request.method == 'POST':
        ...
    else:
        return render_template('index.html')
```



Program *Image Retrieval* Sederhana(8)

Tambahkan kode sebuah fungsi untuk menghasilkan indeks urutan dari sekumpulan citra.

```
def mainImageRetrieval(image_input):
    ENCODER_MODEL = load_model('./static/assets/model/model.h5')
    DATASET_ARRAY =
np.load('./static/assets/array-datasets/dataset_kecil_64x64.npy').astype('float32') / 255
    image_input = image_input.reshape(
        1, image_input.shape[0], image_input.shape[1], image_input.shape[2])
    feature_image_input = ENCODER_MODEL.predict(image_input)
    feature_image_dataset = ENCODER_MODEL.predict(DATASET_ARRAY)
    indices_val_of_sim = checkSimilarity(feature_image_input, feature_image_dataset)
    return indices_val_of_sim
```



Program Image Retrieval Sederhana(9)

• Tambahkan kode sebuah fungsi untuk menghitung kemiripan antara citra kueri dengan citra dataset.

```
def checkSimilarity(input img, dataset img):
    num of database = dataset img.shape[0]
    val of sim = np.array([], dtype='float32')
    # perhitungan kemiripan dengan euclidean distance
    for i in range (num of database):
        selisih antar titik = input img[0] - dataset img[i]
        kuadrat antar titik = selisih antar titik**2
        total semua titik = np.sum(kuadrat antar titik)
        nilai kemiripan = total semua titik**0.5  # akar pangkat 2
        val of sim = np.append(val of sim, [nilai kemiripan])
    val of sim sort = np.sort(val of sim) # pengurutan nilai kemiripan
    indices val of sim = np.argsort(val of sim) # index pengurutan nilai kemiripan
    return indices val of sim
```



Program *Image Retrieval* Sederhana (10)

- Simpan file app.py.
- Buat folder baru dengan nama templates.
- Buat file html baru dengan nama index.html di dalam folder templates.
- Buka file html tersebut dan tambahkan kode awal.



Program *Image Retrieval* Sederhana (11)

- Tambahkan kode untuk menggunakan pustaka Bootstrap dan jQuery.
- Bootstrap: https://getbootstrap.com/
- jQuery: https://api.jquery.com/

```
<head>
...
k rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
...
</head>
```



Program *Image Retrieval* Sederhana (12)

- Tambahkan kode untuk menggunakan pustaka Bootstrap dan jQuery.
- Bootstrap: https://getbootstrap.com/
- jQuery: https://api.jquery.com/



Program *Image Retrieval* Sederhana (13)

Tambahkan kode untuk header.



Program *Image Retrieval* Sederhana (14)

• Tambahkan kode untuk konten dan tempatkan dibawah header.



Program *Image Retrieval* Sederhana (15)

• Tambahkan kode untuk bagian pertama yang akan diisi dengan form input dan citra kueri.



Program *Image Retrieval* Sederhana (16)

• Tambahkan kode untuk form input pada bagian pertama.



Program *Image Retrieval* Sederhana (17)

• Tambahkan kode untuk citra kueri yang akan diunggah.



Program *Image Retrieval* Sederhana (18)

• Tambahkan kode untuk citra kueri yang telah diunggah.

```
<div class="container-fluid" style="margin-top:10px;">
    <div class="row ml-1">
        {% if query img %}
        <div class="col-md-3" style="margin-bottom: 5px;">
            <div class="col-centered">
                {# citra input #}
                <imq class="imq-fluid img-thumbnail" src="{{ url for('static',</pre>
filename='assets/images/upload/') }}{{ query img }}" alt="query image">
            </div>
        </div>
        {% endif %}
    </div>
</div>
```



Program *Image Retrieval* Sederhana (19)

Tambahkan kode untuk hasil sekumpulan citra yang dianggap relevan oleh sistem.

```
<div class="container-fluid" style="margin-top:10px;">
    {% if data %} <div class="row">
        {% for result in data %}
        <div class="col-sm-4 col-md-3">
            <a class="lightbox" href="{{url for('static',</pre>
filename='assets/images/dataset/')}}{{result}}">
            <img class="img-fluid img-thumbnail" style="margin-bottom: 5px;"</pre>
src="{{url for('static', filename='assets/images/dataset/')}}{{result}}" alt="result">
            </a>
        </div> {% endfor %}
    </div>
    {% endif %}
</div>
```



Program *Image Retrieval* Sederhana (20)

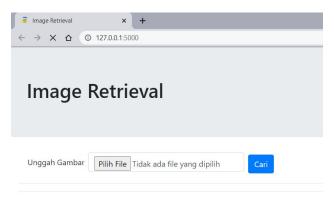
• Tambahkan kode javascript untuk menampilkan citra kueri setelah diunggah.

```
<div class="container-fluid" style="margin-top:10px;">
    <script>
        function readURL(input) {
            if (input.files && input.files[0]) {
                var reader = new FileReader();
                reader.onload = function(e) {
                    $('#blah').attr('src', e.target.result) };
                document.getElementById("blah").style.display = "block";
                reader.readAsDataURL(input.files[0]);
    </script>
</div>
```



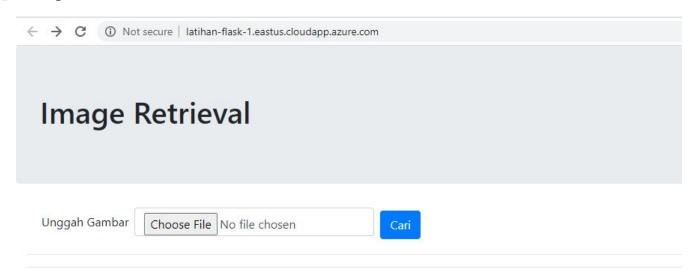
Program *Image Retrieval* Sederhana (21)

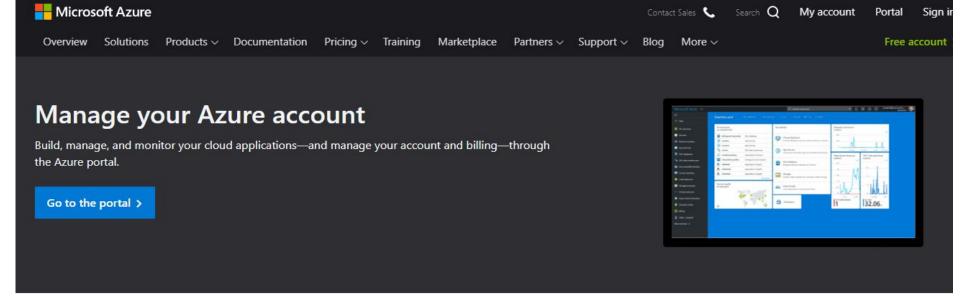
- Simpan file index.html
- Kemudian jalankan kembali file app.py dan buka halaman tersebut di browser.





Deploy on Azure





Stay up to date with your account







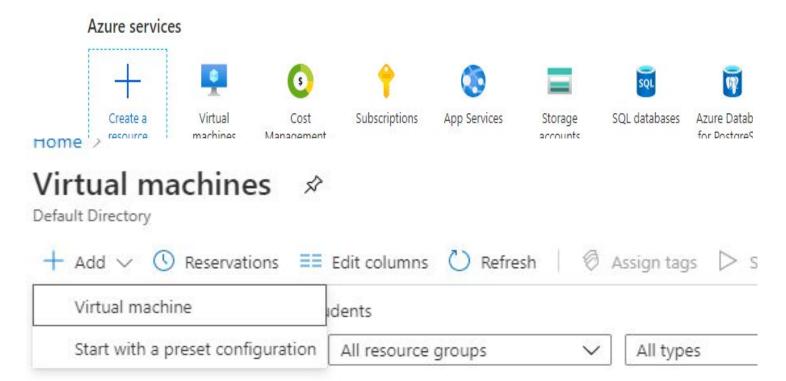
Manage my subscriptions and see usage and billing.

Review my support requests.

Create service health alerts and view service issues.

1. Membuat akun azure for student





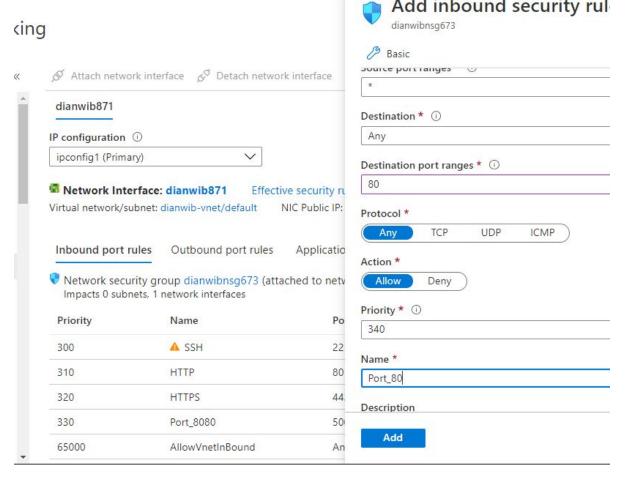
1 items

2. Membuat virtual machine

Home > Virtual machines >	
Create a virtual mad	hine
y	
Subscription * ①	Azure for Students
Resource group * ①	(New) Resource group
	Create new
Instance details	
Virtual machine name * ①	
Region * ①	(US) East US
Availability options ①	No infrastructure redundancy required
Image * ①	Ubuntu Server 18.04 LTS - Gen1
	Browse all public and private images
Azure Spot instance ①	Yes No
Size * ①	Standard_B2s - 2 vcpus, 4 GiB memory (Rp 474.955,52/month) Select size
Administrator account	
Authentication type ①	SSH public key Password
Username * ①	
	The value must not be empty.
	Username must only contain letters, numbers, hyphens, and underscores and may not start with a hyphen or number.
	The value must be between 1 and 64 characters long.
Password * ①	· · · · · · · · · · · · · · · · · · ·
Confirm password * ①	8



3. Setting DNS name



4. Setting permission port

Download PuTTY: latest release (0.74)

Home | FAQ | Feedback | Licence | Updates | Mirrors | Keys | Links | Team Download: Stable · Snapshot | Docs | Changes | Wishlist

This page contains download links for the latest released version of PuTTY. Currently this is 0.74, released on 2020-06-27.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a permanent link to

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code availathen it might be worth trying out the <u>development snapshots</u>, to see if the problem has already been fixed in those versions.

Package files

You probably want one of these. They include versions of all the PuTTY utilities.

(Not sure whether you want the 32-bit or the 64-bit version? Read the FAQ entry.)

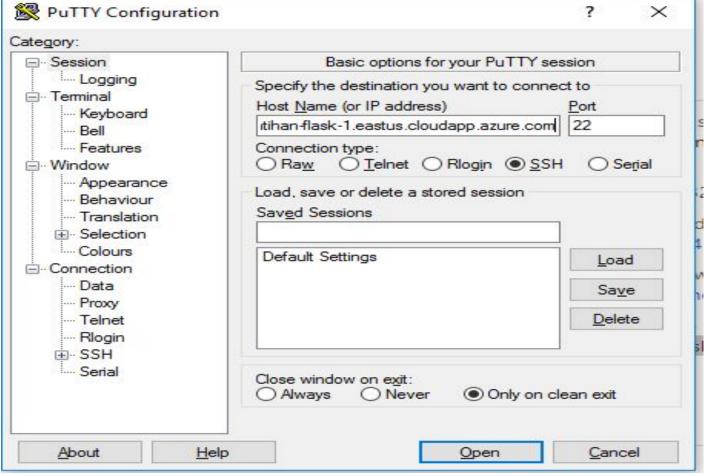
MSI ('Windows Installer')

32-bit: <u>putty-0.74-installer.msi</u> (or by FTP) (signature)
64-bit: <u>putty-64bit-0.74-installer.msi</u> (or by FTP) (signature)

Unix source archive

.tar.gz: putty-0.74.tar.gz (or by FTP) (signature)

5. Download aplikasi "PuTTY"



6. Remote virtual machine by PuTTY

```
Last login: Fri Aug 14 05:32:01 2020 from 111.94.13.251
dianwib@dianwib:~$ sudo su
root@dianwib:/home/dianwib# git clone https://github.com/wahyuakbarwibowo/app-im
age-retrieval.git
```

root@dianwib:/home/dianwib# cd app-image-retrieval/

7. Download file from GitHub

```
root@dianwib:/home/dianwib/app-image-retrieval nano Dockerfile

GNU nano 2.5.3 File: Dockerfile

FROM python:3.8-buster

FROM python:3.6.5-alpine

WORKDIR /project

ADD . /project
```

RUN pip3 install -r requirements.txt

CMD ["python3", "app.py"]

root@dianwib:/home/dianwib/app-image-retrieval# docker image build -t python-fla

docker image build -t python-flask.

```
root@dianwib:/home/dianwib/app-image-retrieval# docker run -p 80:5000 -d python-flask docker run -p 80:5000 -d python-flask
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
RTS	NAMES			***************************************
386a622b306c	python-flask	"python3 app.py"	2 hours ago	Up 2 hours
0.0.0:80->80/tcp	romantic_noyce			

9. Build Docker

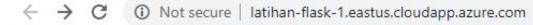


Image Retrieval

Unggah Gambar Choose File No file chosen Cari