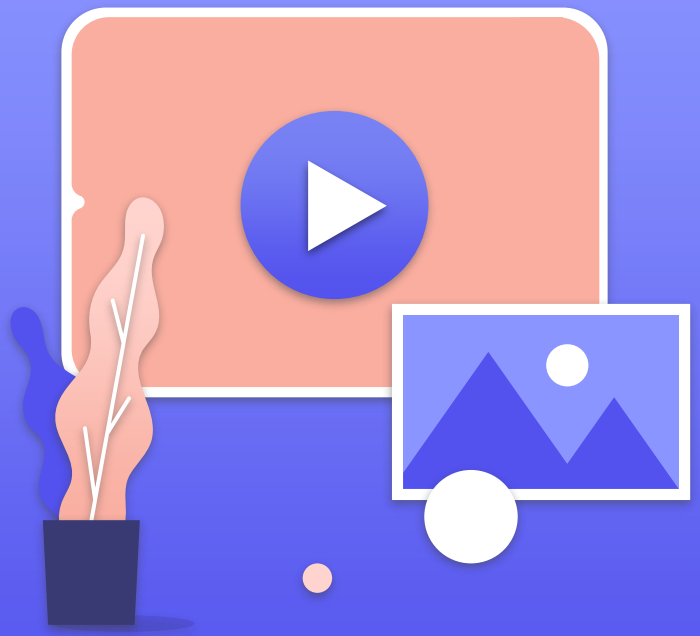


IMAGE RECOGNITION

HANDCRAFT-BASED APPROACH & LEARNING-BASED APPROACH

การรู้จำภาพใบหน้าคน

Outline

- Dataset
- Handcraft-Based Approach
- Learning-Based Approach

. . . . . .

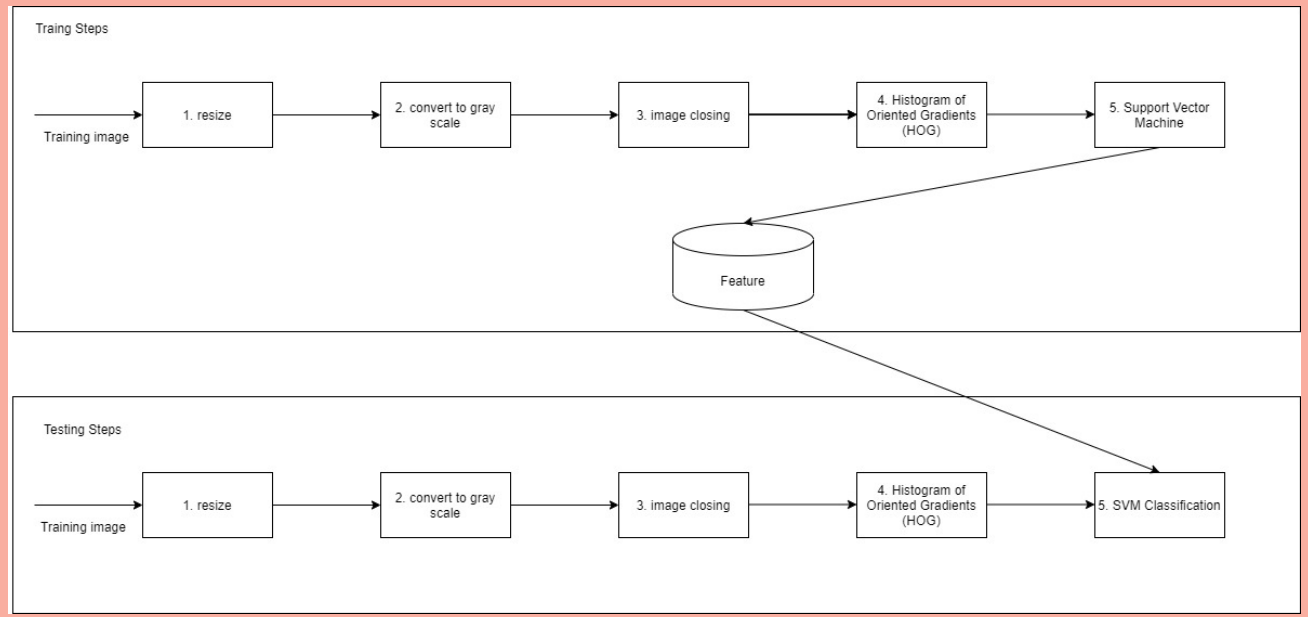
Dataset

Dataset

เรามีวิธีการประมวลผลภาพเพื่อให้คุณลักษณะมีความชัดเจนมากยิ่งขึ้นโดยใช้วิธีการ Closing และตัว Dataset ที่ใช้ได้ทำ Data augmentation ด้วย โดยแต่ละภาพมี Object หลักคือใบหน้าคน

Handcraft-Based Approach

Diagram


Traing Steps
Training image
1. resize
2. convert to gray scale
3. image closing
4. Histogram of Oriented Gradients (HOG)
5. Support Vector Machine
Feature
Testing Steps
Training image
1. resize
2. convert to gray scale
3. image closing
4. Histogram of Oriented Gradients (HOG)
5. SVM Classification

การประเมินคุณลักษณะ

วิธีการประเมินประสิทธิภาพที่เลือกใช้ คือ ซัพพอร์ตเวกเตอร์แมชชีน (SVM) โดยช่วงของการ train ให้ทำการคำนวนผลและฝึกฝนไปก่อน เมื่อนำข้อมูลมาทำการ test ข้อมูลของตัวที่ test จะทำการเรียนรู้จากตัวที่เคยผ่านการาฝึกฝนมากอ่นหน้านี้

Test โดยทำ Image Processing

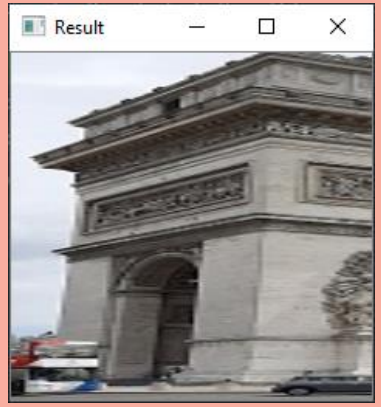
Gray Scale

Data augmentation

Data augmentation

Horizontal shift เลื่อนรูปภาพซ้ายหรือขวาแบบสุ่ม

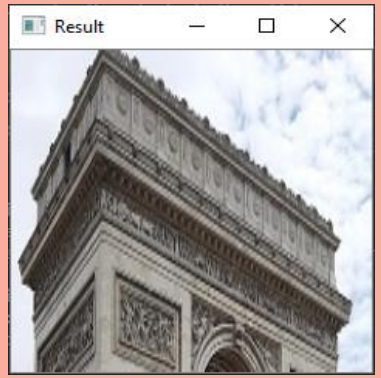
```python
def horizontal_shift(img, ratio=0.0):
    if ratio > 1 or ratio < 0:
        print('Value should be less than 1 and greater than 0')
        return img
    ratio = random.uniform(-ratio, ratio)
    h, w = img.shape[:2]
    to_shift = w * ratio
    if ratio > 0:
        img = img[:, :int(w - to_shift), :]
    if ratio < 0:
        img = img[:, int(-1 * to_shift):, :]
    img = fill(img, h, w)
    return img
```



Data augmentation

Vertical shift เลื่อนรูปภาพบนหรือล่างแบบสุ่ม

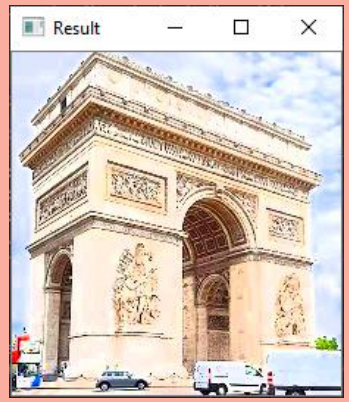
```python
def vertical_shift(img, ratio=0.0):
    if ratio > 1 or ratio < 0:
        print('Value should be less than 1 and greater than 0')
        return img
    ratio = random.uniform(-ratio, ratio)
    h, w = img.shape[:2]
    to_shift = h*ratio
    if ratio > 0:
        img = img[:int(h-to_shift), :, :]
    if ratio < 0:
        img = img[int(-1*to_shift):, :, :]
    img = fill(img, h, w)
    return img
```



Data augmentation

Brightness ลดหรือเพิ่มความสว่างของรูป

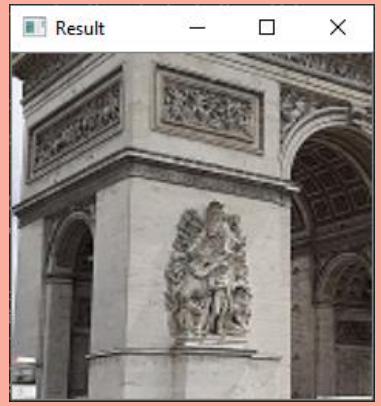
```python
def brightness(img, low, high):
    value = random.uniform(low, high)
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    hsv = np.array(hsv, dtype=np.float64)
    hsv[:, :, 1] = hsv[:, :, 1]*value
    hsv[:, :, 1][hsv[:, :, 1] > 255] = 255
    hsv[:, :, 2] = hsv[:, :, 2]*value
    hsv[:, :, 2][hsv[:, :, 2] > 255] = 255
    hsv = np.array(hsv, dtype=np.uint8)
    img = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)
    return img
```



Data augmentation

Zoom ขยายเข้าหรือออกแบบสุ่ม

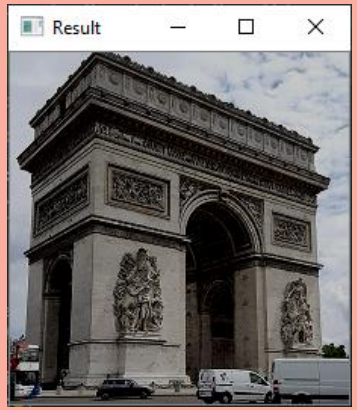
```python
def zoom(img, value):
    if value > 1 or value < 0:
        print('Value for zoom should be less than 1 and greater than 0')
        return img
    value = random.uniform(value, 1)
    h, w = img.shape[:2]
    h_taken = int(value*h)
    w_taken = int(value*w)
    h_start = random.randint(0, h-h_taken)
    w_start = random.randint(0, w-w_taken)
    img = img[h_start:h_start+h_taken, w_start:w_start+w_taken, :]
    img = fill(img, h, w)
    return img
```



Data augmentation

Channel shift สุ่มเปลี่ยนสี

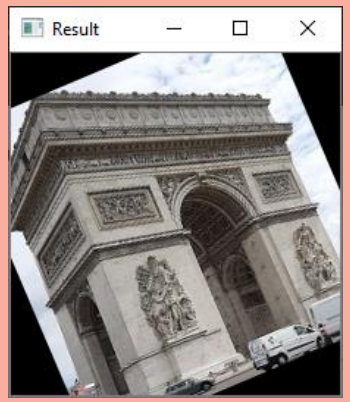
```python
def channel_shift(img, value):
    value = int(random.uniform(-value, value))
    img = img + value
    img[:, :, :][img[:, :, :] > 255] = 255
    img[:, :, :][img[:, :, :] < 0] = 0
    img = img.astype(np.uint8)
    return img
```



Data augmentation

Rotation หมุนภาพแบบสุ่ม

```python
def rotation(img, angle):
    angle = int(random.uniform(-angle, angle))
    h, w = img.shape[:2]
    m = cv2.getRotationMatrix2D((int(w/2), int(h/2)), angle, 1)
    img = cv2.warpAffine(img, m, (w, h))
    return img
```



Closing

การนำ image มา dilate แล้ว ค่อย erode ใช้ในการลบ small holes สามารถใช้ในการ เชื่อมวัตถุที่แยกจากกัน

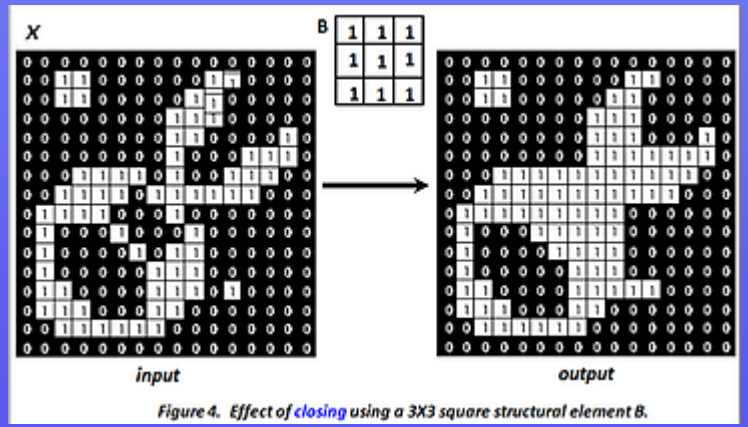


Figure 4. Effect of closing using a 3X3 square structural element B.

• • • • • • •

Result

```
  0%|          | 0/15 [00:00<?, ?it/s]i (1)
Success Rate: 90.00
Failure Rate: 10.00
```

```
 47%|████      | 7/15 [00:03<00:04,  1.81it/s]i (2)
Success Rate: 90.00
Failure Rate: 10.00
```

```
 53%|████      | 8/15 [00:04<00:04,  1.71it/s]i (3)
Success Rate: 90.00
Failure Rate: 10.00
```

```
 60%|█████     | 9/15 [00:04<00:03,  1.73it/s]i (4)
Success Rate: 70.00
Failure Rate: 30.00
```

```
 67%|██████    | 10/15 [00:05<00:02,  1.76it/s]i (5)
Success Rate: 90.91
Failure Rate: 9.09
```

```
 73%|██████    | 11/15 [00:06<00:02,  1.72it/s]i (6)
Success Rate: 90.00
Failure Rate: 10.00
```

```
 80%|███████   | 12/15 [00:06<00:01,  1.79it/s]i (7)
Success Rate: 80.00
Failure Rate: 20.00
```

```
 87%|████████  | 13/15 [00:07<00:01,  1.88it/s]i (8)
Success Rate: 100.00
Failure Rate: 0.00
```

Result

```
93%|█████████ | 14/15 [00:07<00:00,  1.87it/s]i (9)
Success Rate: 70.00
Failure Rate: 30.00
```

```
27%|███       | 4/15 [00:01<00:05,  2.06it/s]i (13)
Success Rate: 30.00
Failure Rate: 70.00
```

```
 7%|▋         | 1/15 [00:00<00:06,  2.01it/s]i (10)
Success Rate: 80.00
Failure Rate: 20.00
```

```
33%|███       | 5/15 [00:02<00:05,  1.88it/s]i (14)
Success Rate: 70.00
Failure Rate: 30.00
```

```
13%|█▎        | 2/15 [00:00<00:06,  2.01it/s]i (11)
Success Rate: 80.00
Failure Rate: 20.00
```

```
40%|████      | 6/15 [00:03<00:04,  1.85it/s]i (15)
Success Rate: 90.00
Failure Rate: 10.00
```

```
20%|██        | 3/15 [00:01<00:05,  2.02it/s]i (12)
Success Rate: 90.00
Failure Rate: 10.00
```

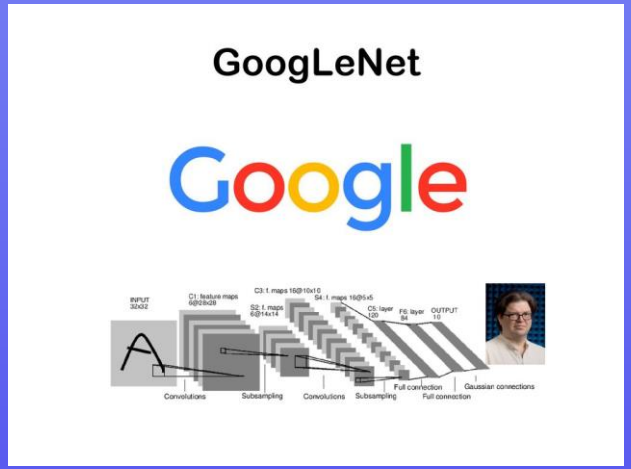
Learning-Based Approach

GoogLeNet

ใน Project นี้เราเลือกใช้ structure คือ GoogLeNet

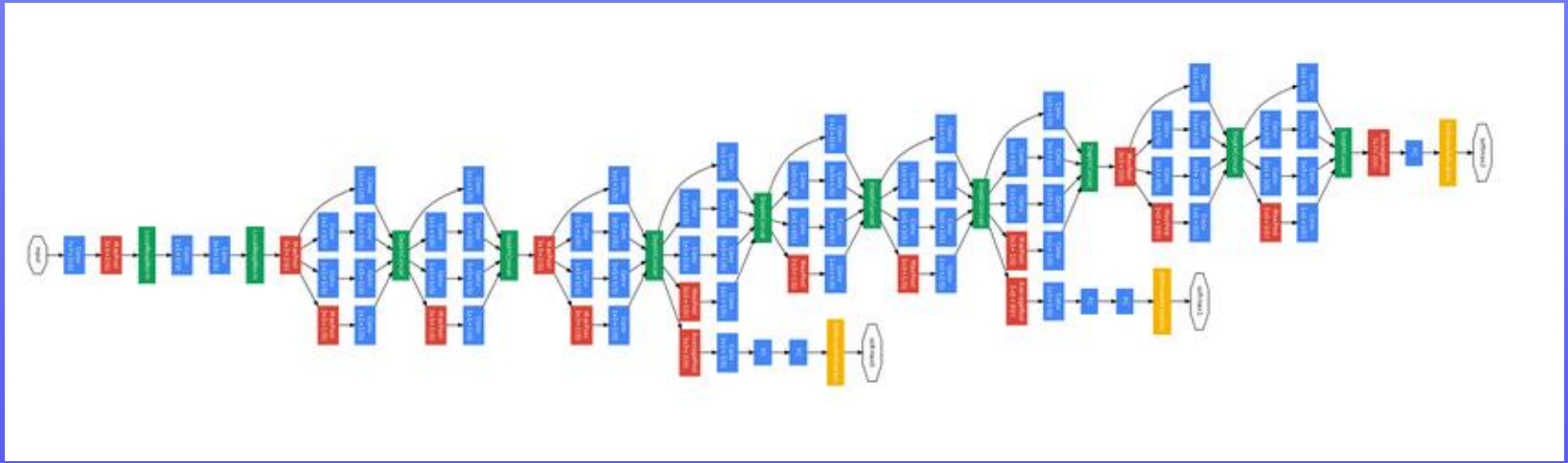
โดย GoogLeNet เป็นโครงข่ายประสาทเทียมแบบ deep convolutional 22 ชั้นซึ่งเป็นรูปแบบหนึ่งของ Inception Network

GoogLeNet ประกอบด้วย 22 เลเยอร์ (27 เลเยอร์รวมเลเยอร์รวมกัน) และส่วนหนึ่งของเลเยอร์เหล่านี้เป็นโมดูลเริ่มต้นทั้งหมด 9 โมดูล GoogLeNet เป็นสถาปัตยกรรมหลักภายในไลบรารี ML ทั่วไป



Structure

Structure



Data augmentation

Data augmentation

```
PRETRAINED_SIZE = 224  # define pretrained size
PRETRAINED_MEANS = [0.485, 0.456, 0.406]  # define pretrained means
PRETRAINED_STDS = [0.229, 0.224, 0.225]  # define pretrained stds
TRAIN_TRANSFORMS = transforms.Compose([  # define transforms for train datasets
        transforms.Resize(PRETRAINED_SIZE),
        transforms.RandomRotation(5),
        transforms.RandomHorizontalFlip(0.5),
        transforms.RandomCrop(PRETRAINED_SIZE, padding=10),
        transforms.RandomVerticalFlip(0.5),
        transforms.RandomRotation(random.randint(0, 360)),
        transforms.RandomGrayscale(0.1),
        #transforms.RandomErasing(0.5),
        transforms.ToTensor(),
        transforms.Normalize(mean=PRETRAINED_MEANS, std=PRETRAINED_STDS)])
```

```
TEST_TRANSFORMS = transforms.Compose([  # define transforms for test datasets
        transforms.Resize(PRETRAINED_SIZE),
        #transforms.CenterCrop(PRETRAINED_SIZE),
        transforms.ToTensor(),
        transforms.Normalize(mean=PRETRAINED_MEANS, std=PRETRAINED_STDS)])
BATCH_SIZE = 40  # define batch size
```

Result

```
Accuracy for class i (1) is: 60.0 %
Accuracy for class i (10) is: 70.0 %
Accuracy for class i (11) is: 90.0 %
Accuracy for class i (12) is: 60.0 %
Accuracy for class i (13) is: 30.0 %
Accuracy for class i (14) is: 80.0 %
Accuracy for class i (15) is: 60.0 %
Accuracy for class i (2) is: 60.0 %
Accuracy for class i (3) is: 80.0 %
Accuracy for class i (4) is: 50.0 %
Accuracy for class i (5) is: 90.9 %
```

```
Accuracy for class i (6) is: 70.0 %
Accuracy for class i (7) is: 70.0 %
Accuracy for class i (8) is: 70.0 %
Accuracy for class i (9) is: 60.0 %
```

สมาชิก

นายชวิน โล่ห์รัตนเสน่ห์

62070045

นายณัฐวัฒน์ สามสี

62070067

นายเดชพนต์ นุ่นเสน

62070070

นายธีรภัทร์ บุญช่วยแล้ว

62070096