



ຄະນະ ວິທະຍາສາດທຳມະຊາດ
ສາຂາ ວິທະສາສາດຄອມພິວເຕີ
ວິຊາ: ການຊຽນໂປຣມແກຣມດ້ວຍພາສາ Python
Project: Mask Detection

ສອນໂດຍ: ອຈ ປອ ສົມສັກ ອິນທະສອນ

ນຳສະເໜີໂດຍ: 1. ທ. ປິທຸ ກາປໍ ຫ້ອງ 1CS2

2. ທ. ທິວິຊິງ ດົງບູນ ຫ້ອງ 1CS2

3. ທ. ປໍວີ ປາງຈາວວີ ຫ້ອງ 1CS2

2020-2021

1. ຄວາມໝາຍ

Mask detection ແມ່ນໂປຣແກຣມໜຶ່ງທີ່ໃຊ້ໃນການກວດສອບແມັດ ຂອງບຸກຄົນທີ່ໃສ່ mask ຫຼື ບໍ່ໃສ່ mask.

2. ຄວາມສຳຄັນ

Mask detection ມີຄວາມສຳຄັນຫຼາຍໃນປັດຈຸບັນ ໂດຍສະເພາະໃນຊ່ວງວິກິດທີ່ ພະຍາດ COVID-19 ກຳລັງລະບາດໃນສັງຄົມ ເພື່ອໃຫ້ ສະດວກ ແລະ ຄອບຄຸມໄດ້ດີໃນການເຂົ້າອອກຂອງຄົນເຮົາ

3. ຈຸດປະສົງ

ສ້າງໂຄດນີ້ຂຶ້ນເພື່ອ:

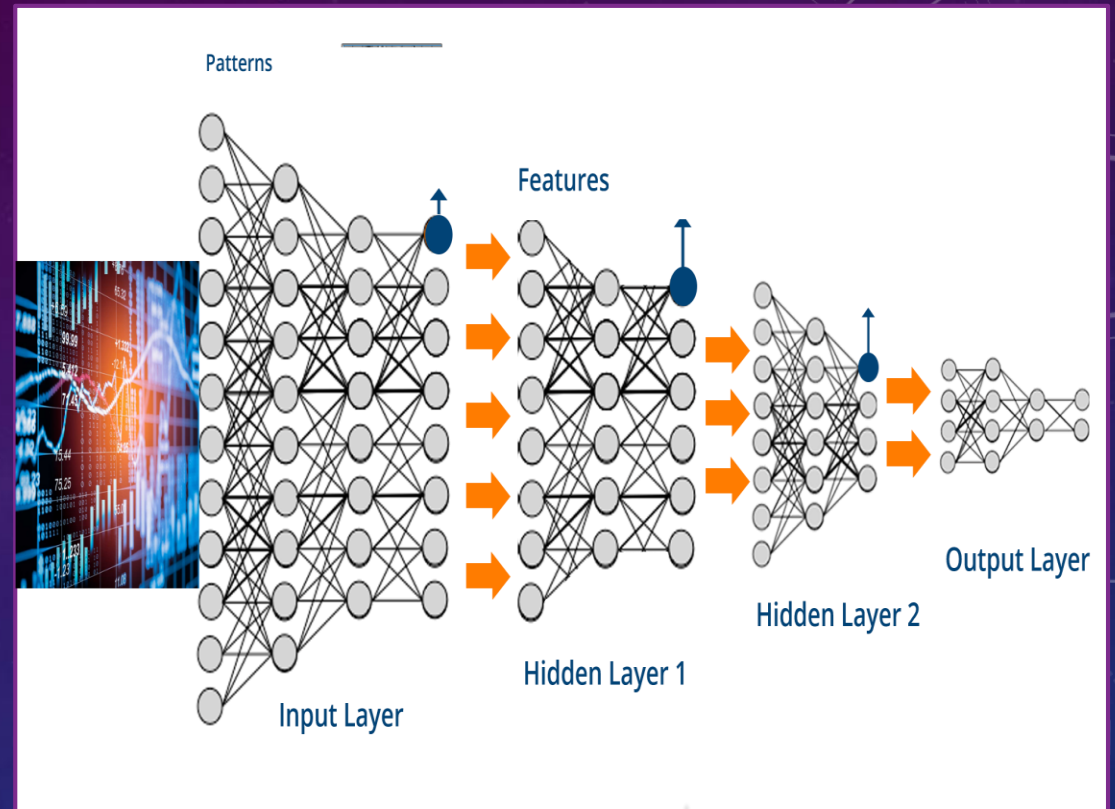
- ເພື່ອກວດຫາຄົນທີ່ບໍ່ໃສ່ Mask
- ເພື່ອຫຼຸດໄລຍະເວລາໃນການກວດ Mask ຕາມສະຖານທີ່ຕ່າງໆ
- ເພື່ອຫຼຸດແຮງງານຈາກຄົນທີ່ຕ້ອງກວດ Mask

4. ຜົນທີ່ຄາດວ່າຈະໄດ້ຮັບ:

- ສາມາດທີ່ຈະນຳເອົາລະບົບໄປຕໍ່ຍອດນິຍົມຫຼືປະຍຸກໃນການພັດທະນາຕໍ່ໄດ້
- ຫຼຸດໄລຍະເວລາໃນການກວດຈັບ Mask ໃນໄລຍະທີ່ມີບຸກຄົນຈຳນວນຫຼາຍຢູ່
- ສາມາດຮຽນຮູ້ແລະເຂົ້າໃຈໃນເລື່ອງຂອງການປະມວນຜົນຮູບພາບ
- ຄັດແຍກບຸກຄະລິກກະພາບໃສ່ Mask ແລະບໍ່ໃສ່ Mask ໄດ້ຢ່າງໄວ

❖ Tensorflow ແມ່ນຫຍັງ?

Tensorflow ເປັນ Library ແບບ Open Source ສໍາລັບການໃຊ້ພັດທະນາ Machine learning ຖືກພັດທະນາຂຶ້ນມາໂດຍ Google ໂດຍ Google ໄດ້ປ່ອຍ Library ທີ່ໃຊ້ງານກັນຢູ່ໃນ Google ເອງໃຫ້ກາຍເປັນແບບ Open Source ໃຫ້ທຸກຄົນສາມາດເຮັດໄດ້ທີ່ຈະນໍາເອົາມາໃຊ້ໄດ້



❖ NumPy ເປັນຊື່ຂອງ library ທີ່ໃຊ້ໃນການຄຳນວນທາງຄະນິດສາດ ແລະ ມີຄວາມສາມາດໃນການຈັດການກັບ Array ຫຼາຍມິຕິ

```
>>> a[0, 3:5]
array([3, 4])

>>> a[4:, 4:]
array([[44, 55],
       [54, 55]])

>>> a[:, 2]
a([2, 12, 22, 32, 42, 52])

>>> a[2::2, ::2]
array([[20, 22, 24],
       [40, 42, 44]])
```

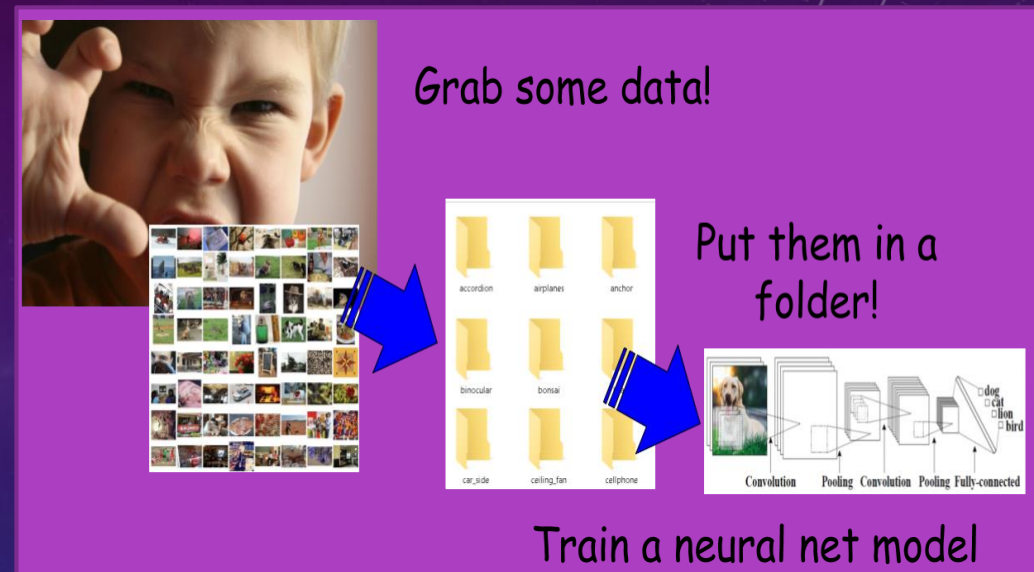
0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

❖ Keras ແມ່ນຫຍັງ?

Keras ແມ່ນ deep learning framework ສໍາລັບ python ທີ່ທຳໃຫ້ວິທີຂອງການ train model ດ້ວຍ deep learning

❖ Pillow ແມ່ນຫຍັງ?

ແມ່ນຕົວທີ່ເຮົາໃຊ້ດຶງຮູບພາບແລະເປັນຕົວແປງຮູບພາບ



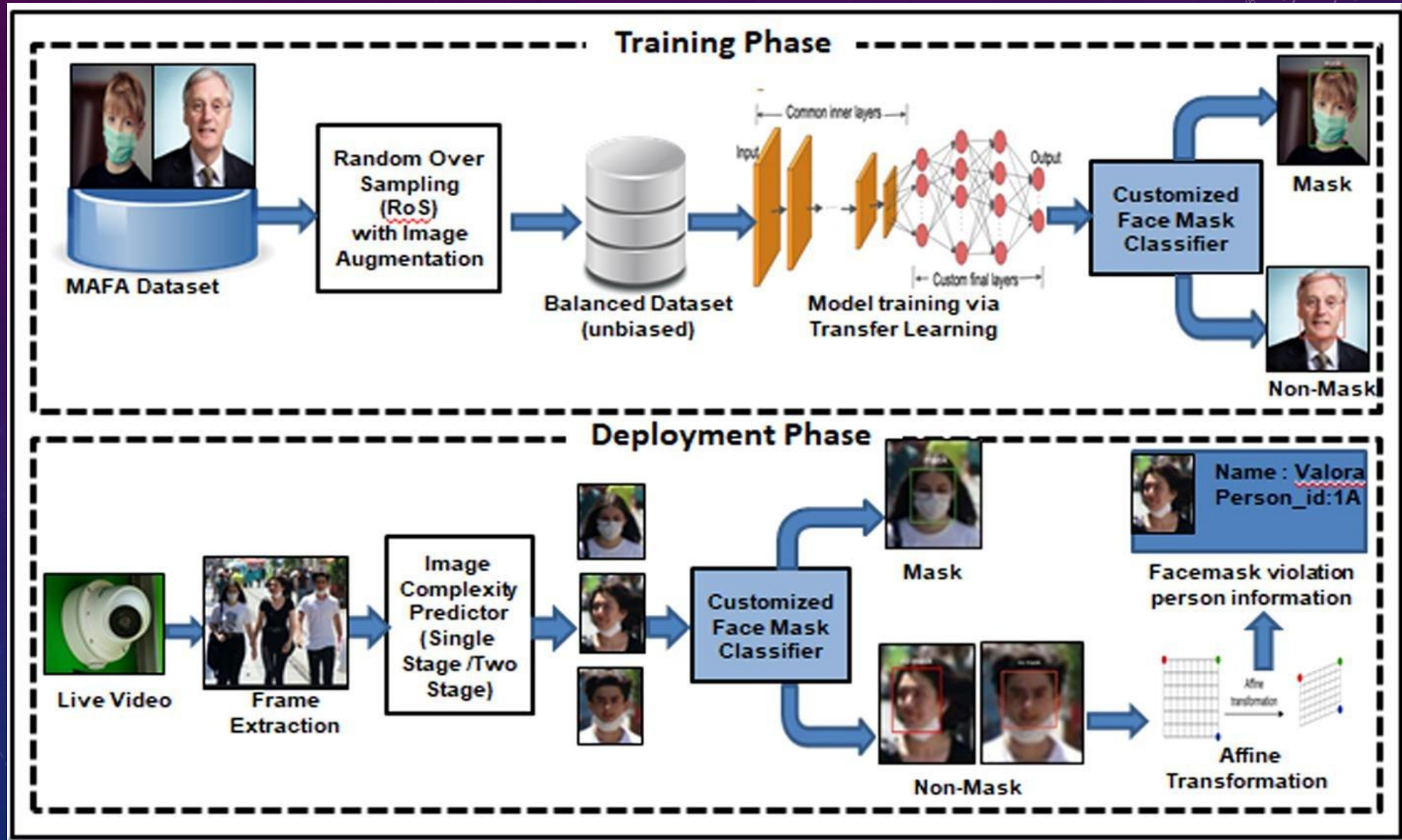
□ ສ້າງ Classification Model

Teachable Machine ທີ່ຈະຮັບຮູບພາບເພື່ອໃຫ້ Model ລະບຸພາບທີ່ຕ້ອງການໄດ້ ເພື່ອແບ່ງປະເພດພາບທີ່ເຫັນອອກເປັນປະເພດຕ່າງໆ ເຊັ່ນ: ຄົນໃສ່ mask ແລະ ຄົນບໍ່ໃສ່ Mask ເຊິ່ງຈະແບ່ງຂໍ້ມູນອອກເປັນຈັກປະເພດກໍໄດ້ ເຊິ່ງເອີ້ນວ່າ Class ເຮົາຈະເອົານຳພາບຂອງແຕ່ລະ Class ເຂົ້າໄປໃຫ້ Machine Learning ຮຽນຮູ້ ແລະ ດາວໂຫລດໂປຣແກຣມທີ່ True ສຳເລັດແລ້ວມາໃຊ້ງານສອນໜຶ່ງຂອງໂຄດຂອງເຮົາ

□ ສ້າງ Model ດ້ວຍໂປຣແກຣມສຳເລັດຮູບ

<https://teachablemachine.withgoogle.com/train/image>

□ ຮູບແບບຂອງການ ສ້າງ Model:



□ ທຳຄວາມເຂົ້າໃຈ Model ເບື້ອງຕົ້ນ:

```
In [9]: from keras.models import load_model
        from PIL import Image, ImageOps
        import numpy as np

        # Load the model
        model = load_model('keras_model.h5')

        # Create the array of the right shape to feed into the keras model
        # The 'Length' or number of images you can put into the array is
        # determined by the first position in the shape tuple, in this case 1.
        data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
        # Replace this with the path to your image
        image = Image.open('mask_3.jpg')
        #resize the image to a 224x224 with the same strategy as in TM2:
        #resizing the image to be at least 224x224 and then cropping from the center
        size = (224, 224)
        image = ImageOps.fit(image, size, Image.ANTIALIAS)

        #turn the image into a numpy array
        image_array = np.asarray(image)
        # Normalize the image
        normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1
        # Load the image into the array
        data[0] = normalized_image_array

        # run the inference
        prediction = model.predict(data)
        print(prediction)
```


□ ນຳເອົາ Model ມາປະກອບກັບ webcam ແລະ Face Detection:

```
In [1]: import cv2
import tensorflow
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np

webcam = cv2.VideoCapture(0)
face_cascade = 'haarcascade_frontalface_default.xml'
face_classifier = cv2.CascadeClassifier(face_cascade)
model = load_model('keras_model.h5')
size = (224, 224)
while True:
    success, image_bgr = webcam.read()
    if success:
        image_bw = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
        image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
        faces = face_classifier.detectMultiScale(image_bw)
        for face in faces:
            x, y, w, h = face
            cface_rgb = Image.fromarray(image_rgb[y:y+h,x:x+w])
            data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
            image = cface_rgb
            image = ImageOps.fit(image, size, Image.ANTIALIAS)
            image_array = np.asarray(image)
            normalized_image_array = (image_array.astype(np.float32) / 127.0) - 1
            data[0] = normalized_image_array
            prediction = model.predict(data)
            print(prediction)
```

```
    if prediction[0][0] < prediction[0][1]:
        cv2.putText(image_bgr, 'MASKED', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0,255,0), 2)
    else:
        cv2.putText(image_bgr, 'NON_MASKED', (x, y-7), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 0, 255), 2)
        cv2.rectangle(image_bgr, (x, y), (x+w, y+h), (0,0,255), 2)
cv2.imshow("Mask Detection", image_bgr)
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
else:
    break
webcam.release()
cv2.destroyAllWindows()
```

ຂອບໃຈ