

Face Mask Detection (FMD)

Agostinho Américo, Houeix Julien, Teixeira Daniel

Background

- Une solution pour diminuer les transmissions des virus respiratoires (Influenza, RSV) et actuellement SARS-CoV-2 passe par le port du **masque** de façon **correcte**
- Une solution de feedback en temps réel pourrait aider à **sensibiliser** les personnes à porter le masque correctement
- Le **deep learning** a évolué énormément ces dernières années et la détection d'un visage et d'un masque devient une tâche facilement réalisable par une machine.

Objectif

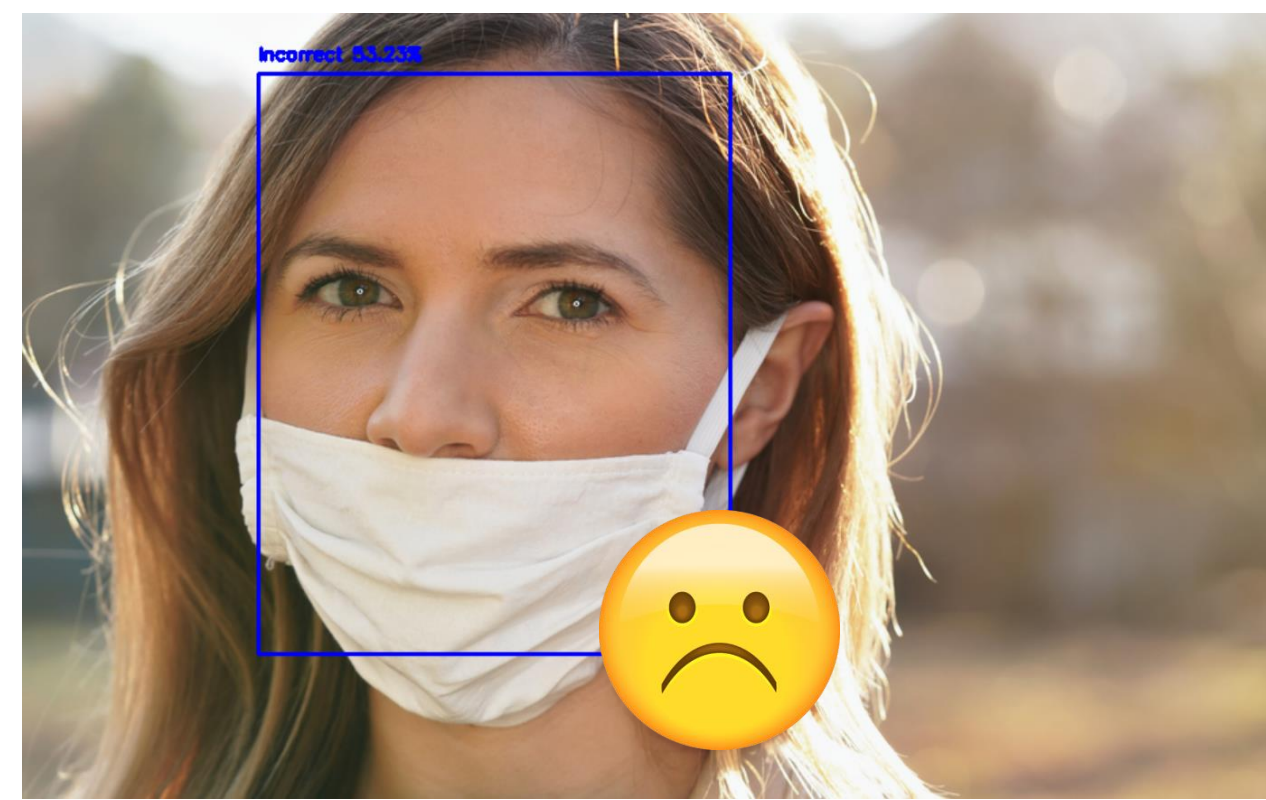
- **Donner un feedback** aux personnes du port correct du masque

Sans masque



Probabilité 100%

Port du masque incorrect



Probabilité 53%

Correct



Probabilité 100%

Méthodologie

- Utilisation comme base de l'algorithme le travail proposé par Adrian Rosebrock [1]
- Entraînement du modèle sur de vrais images de personnes portant des masques [2]
- Entraînement du modèle sur un jeu d'images de masques mal portés artificiellement construit [3]
- Adaptation du réseau de neurones afin de classifier 3 classes au lieu des 2 classes proposées initialement par [1]

Deep / Transfer learning avec Keras

```
#Transfer learning
baseModel = MobileNetV2(weights='imagenet', include_top=False, input_tensor=Input(shape=(224, 224, 3)))

# construct the head of the model that will be placed on top of the the base model
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name='flatten')(headModel)
headModel = Dense(128, activation='relu')(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(3, activation='softmax')(headModel)

# place the head FC model on top of the base model (this will become
# the actual model we will train)
model = Model(inputs=baseModel.input, outputs=headModel)

# loop over all layers in the base model and freeze them so they will
# not be updated during the first training process
for layer in baseModel.layers:
    layer.trainable = False

# compile our model
print("[INFO] compiling model...")
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss=tf.keras.losses.SparseCategoricalCrossentropy(), optimizer=opt, metrics=["accuracy"])
```

Résultats

- Lorsque la personne se trouve en face de la caméra et avec une bonne luminosité la détection du port du masque correct et la détection d'une personne sans masque est **bonne** (testé avec Microsoft surface et pc portable)
- La couleur du masque ou motifs n'ont pas beaucoup d'impacts sur la détection du port du masque
- Quand le masque est mal porté l'algorithme a plus de difficultés à l'interpréter.

Conclusions

- L'algorithme de «masque incorrect» devrait encore être amélioré.
- D'autres senseurs pourraient être connectés comme par exemple un senseur de température.
- Possibilité d'utilisation pour calcul de flux de personnes
- Les moyens nécessaires à ce projet sont simples et peu coûteux.

Références

1. Adrian Rosebrock, Face Alignment with OpenCV and Python (pyimagesearch): OpenCV, Keras, TensorFlow, MobileNetV2
2. Jeux d'images de masques bien portés: www.github.com/chandrikadeb7/Face-Mask-Detection
3. Jeux d'images de masques mal portés: www.github.com/cabani/MaskedFace-Net

Download poster

