(FPT) Data-centric competition

Data-centric competition Andrew Ng

Link competition: https://https-deeplearning-ai.github.io/data-centric-comp/

5th solution (Lots of incredible techniques)

Link: https://towardsdatascience.com/explaining-how-i-reached-the-top-ranks-of-the-new-data-centric-competition-888fc8e86547

Git: https://github.com/pierrelouisbescond/data-centric-challenge-public

The most innovative award of the competition

Link: https://towardsdatascience.com/how-i-won-andrew-ngs-very-first-data-centric-ai-competition-e02001268bda

Augmentation EXPERIMENTS

Remarks

- Shear & rotation might not improve the result of model
- The reason why using Shear and Rotation is not effective is because of the blacking padding after augmentation. All the augmentation data have the black pad around the images, while in the public test there is not —> "padding" is a huge problem
- (**IMPORTANT**!!) After possible running data augmentation code, before doing the experiment, we NEED to visualize all the data in the "public_test" folder to figure out the idea of data augmentation
- On the prediction of the validation set, the model is efficient in predicting the people with a mask but failed to classify the people with no mask or incorrect mask —> DATASET NOT BALANCE (**Need to design a balanced dataset**!!!!!!!!!)
-
- We ***should design the public and validation dataset to contain all the concepts that appeared in the data provided to us by competition*** (!!!!!!!)

Experiments

▼ **EXPERIMENT 1 (DONE)**

- 20% mosaic
- Another is for random_affine() & hsv augmentation

(!!!!)There's a problem with bounding boxes labels after rotating and shearing, still have padding!!!!

▼ **EXPERIMENT 2 (DONE)**

- 70% mosaic augmentation
- The other 30% should use padding to resize the image matching the input of the model

—> Result is also not promising

▼ **EXPERIMENT 3 (DONE)**

- Augment 2000 using only mosaic —> Will hurt the model (B.c # of normal images are just 800) —> model only learns to detect mosaic image type

▼ **EXPERIMENT 4 (DONE)**

- Duplicate the original images —> 749 * 2 = 1498
- Triple the number of public_test images = 187 * 3 = 561

- Mosaic only 300 images

—> Total images = 1498 + 561 + 300 = 2359 images

▼ **EXPERIMENT 5 (DONE)**

Use **Albumentation** library https://github.com/albumentations-team/albumentations#i-want-to-explore-augmentations-and-see-albumentations-in-action

- Random Zoom/scale the images (`RandomScale`)

https://albumentations.ai/docs/api_reference/augmentations/geometric/resize/#albumentations.augmentations.geometric.resize.F

- Gray image color (Done)
- Bend the images from the above images to the center images
- Normalize image for faster training (`Normalize`)

https://albumentations.ai/docs/api_reference/augmentations/transforms/#albumentations.augmentations.transforms.Normalize

- Horizontal flip (`HorizontalFlip`)

https://albumentations.ai/docs/api_reference/augmentations/transforms/#albumentations.augmentations.transforms.HorizontalFli

- Crop a random part of the input and rescale it to some size without loss of bboxes. (`RandomSizedBBoxSafeCrop`)

https://albumentations.ai/docs/api_reference/augmentations/crops/transforms/#albumentations.augmentations.crops.transforms.

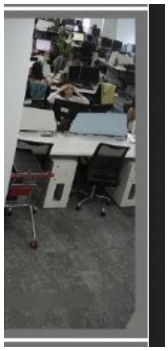
▼ **Experiment 6 (Total 2200 augmented images)**

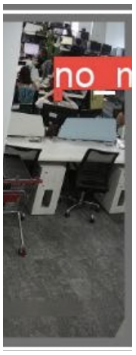
- ToGray (300 images)
- motionBlur (400 images)
- HorizontalFlip (300 images)
- RandomToneCurve (500 images)
- Mosaic (500 images)
- sharpen (200 images)

—> **Score = 0.51** (on Public test)

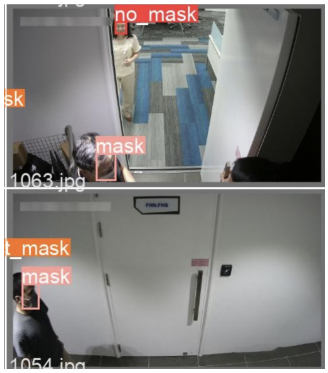
▼ **Remark**

- Models still can not predict the skew face (Non-mask & incorrect_mask types) in the images —> Need more augmentations technique on those images (lots of skew undetected faces)
- Skew face with incorrect masks —> All the skew face with the mask only cover nearly half of the face will be considered as "Incorrect mask" —> Many predictions considered this case as "Mask"
- Some images with these below concept fail to detect face —> Need to augment more images with this concept



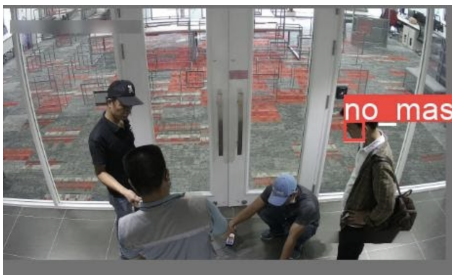


- Images contained faces closed to the their edge fails to detect faces



- Cases that faces are small and stands behind the doors —> Fail to detect them

- Fails to detect people wearing a cap b.c this cases, the face contains a lot of dark area



▼ **Experiment 7 (2000 augmented images)**

- ToGray (200 images)
- motionBlur (500 images)
- HorizontalFlip (500 images)
- RandomToneCurve (300 images)
- Mosaic (500 images)

—> Score on Public set = mAP@.5 = 0.62 (Why????)

—> Score on 40% of private set = mAP@.5 = 0.357 (:<<)

▼ **Remark**

- The values in Public_test don't properly represent the score in 40% private set (WHY??)
- The old validation set successfully represents the data from 40% private test (val's score nearly the same as private test's score)

—> Try labeling based on validation dataset!!! (NO, it does NOT works like it)

- We need to find the right concept of "incorrect mask" —> Find them in "val" and "public_test" dataset

▼ **Experiment 8 (2000 augmented images)**

- ToGray (200 images)
- motionBlur (500 images)
- HorizontalFlip (300 images)

- RandomToneCurve (300 images)
- Mosaic (700 images)

▼ **Data origin experiment 1**

- Path dataset **"Data_competition/dataset_storage/dataset_origin_ex1"**

—> Score on Public set = mAP@.5 = 0.579

—> Score on 40% of private set = mAP@.5 = 0.44 (Submit file name: **submit_0579_best.zip**)

▼ **Remark**

- The result on the private set (0.444) is quite low compared to the Public_test set (0.579)

▼ **Data origin experiment 2 (based on public_test)**

- Path dataset "**Data_competition/dataset_origin**"

—> Score on Public set = mAP@.5 = 0.44

—> Score on 40% of private set = mAP@.5 = (submit name: **submit_pubT_044.zip**)

TODO lists

- [x] ~~Relabel the validation dataset~~
  - [x] ~~Also, list all the concepts of a validation dataset~~
  - [x] ~~Review the validation dataset relabel~~
  - [x] ~~Test on the new validation dataset~~
- [x] ~~Relabel "train" & "Val" set again based on the concept of public_test~~
- [x] ~~**GOAL**: Construct again the public_test & Val dataset so that they cover all the concepts that appeared in the given dataset from the competition (!!!!!)~~
  - [x] ~~Improve KNN or any techniques to separate between different types of image's concepts~~

Questions

- What is the label for HSV augmentation function??
- We can utilize KNN to find the 5 closest images with each image in the public test set from the training dataset (Worth trying!!!!)

> https://github.com/shilparai/image_classification_knn

—> It'll help to generate the training set which is the most appropriate with the public test set

Resources

1. Data augmentation with bounding boxes

https://google.github.io/mediapipe/solutions/face_detection.html

2. Different types of bounding boxes

https://albumentations.ai/docs/getting_started/bounding_boxes_augmentation/

3. Use Wandb to watch model for training YOLOv5

https://wandb.ai/harrypham123/data_FPT_visualization

4. Incorrect/correct masks wearing dataset Sources

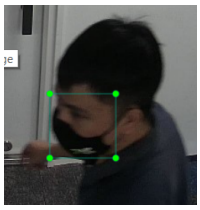
https://www.kaggle.com/andrewmvd/face-mask-detection

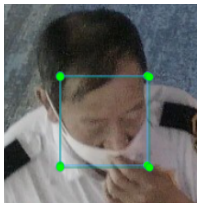
(Create a face mask detection notebook after this competition on Kaggle)

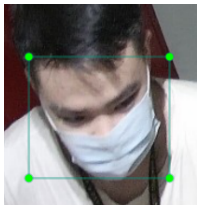
Incorrect_mask analysis

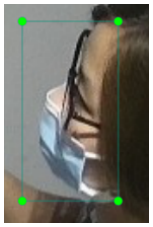
Val

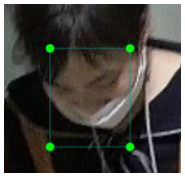
▼ **Suspicious mask**

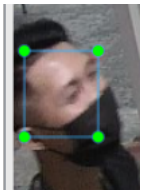


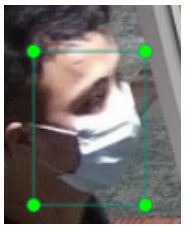


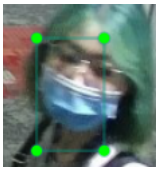


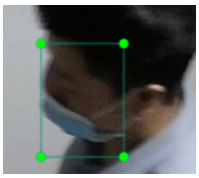




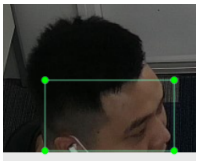


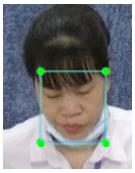


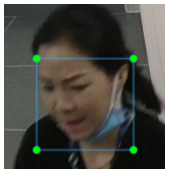


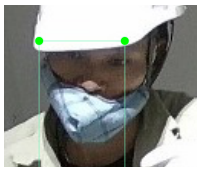


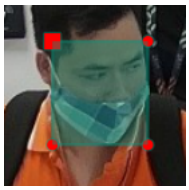
▼ **Incorrect mask**





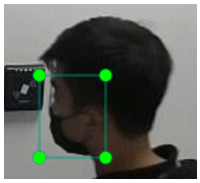


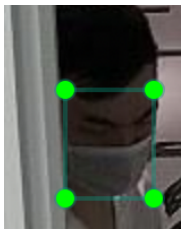


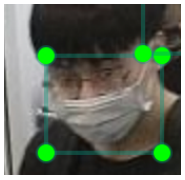


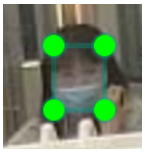
public_test

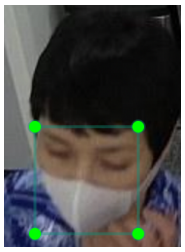
▼ **Suspicious mask**



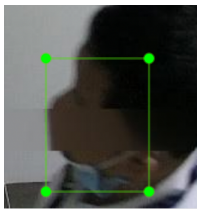


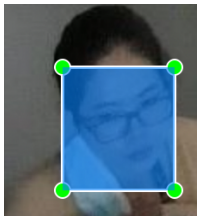






▼ **Incorrect mask**





▼ **Conclusion**

- "**Incorrect_mask**" happens when the mask doesn't cover 2 holes of the nose. In case, mask only cover half of the nose → We also consider them as "**mask**"

Incorrect_mask vs mask analysis

▼ **Remark**

- Because the competition only asks us to submit train & Val dataset —> It's meaningless to do the tricks on public_test set

—> **Public_test** set can be considered as the **CORRECT REFERENCE set —> Label all the data based on this public_test**

▼ **Experiment 1**

> "**Correct mask**" should cover 2/3 part above the nose

☑ ~~All types of "suspicious masks" are considered as "incorrect_mask" (DONE)~~

- "Val" dataset is also relabeled based on the above idea

—> Data is submitted with experiment 8 augmentation above

▼ **Experiment 2**

> **"Correct mask"** only need to cover or hide 2 holes of the nose

- Analysis more closely the concept of "incorrect_mask" in public_test set

—> "**Incorrect_mask**" happens when the mask doesn't cover 2 holes of the nose. In case, mask only cover half of the nose → We also consider them as "**mask**"

- [x] ~~Relabel "Val" dataset based on the concept of public_test~~
- [x] ~~Relabel "train" dataset based on the concept of public_test~~

▼ **TODO (17 Nov)**

- [x] ~~Design the original dataset for Experiment 2~~
- [x] ~~Apply experiment 8 above to see what happen with our relabeled data based on public_test (DONE)~~

▼ **Remark**

The majority of wrong labels comes from the labels "incorrect_mask" & "mask"

—> B.c Data is not balanced, # of Incorrect_mask labels is small while the # of mask labels is large

- [x] ~~Visualize the distributions of each label (mask, non-mask, incorrect mask)~~

→ From visualizing, data is NOT balance

- [x] ~~Write a function to delete all the files with a specific pattern's name from a given folder's path~~
- [x] ~~Write an algorithm to filter only images with the "incorrect_mask" label~~
- [ ] Augment images with "incorrect_mask" to increase their number of labels to get the balanced dataset (!!!!!!)
  - [ ] What type of augmentation is possible for "incorrect_mask" labels

TODO list tracking works

WEEK 4 (15/11 → 21/11)

- [x] ~~Relabel "train" & "Val" set again based on the concept of public_test~~
- [x] ~~**GOAL**: Construct again the public_test & Val dataset so that they cover all the concepts that appeared in the given dataset from the competition (!!!!!)~~
  - [x] ~~Improve KNN or any techniques to separate between different types of image's concepts~~
- [x] ~~Visualize the distributions of each label (mask, non-mask, incorrect mask)~~

→ From visualizing, data is NOT balance

- [x] ~~Write a function to delete all the files with a specific pattern's name from a given folder's path~~
- [x] ~~Write an algorithm to filter only images with the "incorrect_mask" label~~
- [x] ~~Augment images with "incorrect_mask" to increase their number of labels to get the balanced dataset (!!!!!!)~~
  - [x] ~~What type of augmentation is possible for "incorrect_mask" labels~~
- [x] ~~From mosaic augmented data, filter and remove the mosaic images not contained bounding box (DONE)~~
- [x] ~~Improve our code constructions~~

Kaggle great reference: https://www.kaggle.com/sreevishnudamodaran/effdet-pytorch-cutmix-mixup-kfold-cosanneal

▼ **Experiment 9 (2000 augmented images)**

▼ **Annotations**

- non_mask: n_m
- mask: m
- incorrect_mask: i_m

▼ **Label distributions in "train"**

- Non-mask (~ 400) —> Need at least 100+ n_m images
- Mask (~ 850)
- incorrect_mask (~ 100) —> Need at least 300+ i_m images

- ToGray (200 images) (done)
- GaussianBlur or medianBlur (100 i_m images + 100 n_m images + 100 normal images) (done)
- Cutout (200 images + 100 i_m images) (done)
- HorizontalFlip (100 i_m images + 200 normal images) (done)
- HueSaturationValue or RandomBrightnessContrast (300 images) (done)
- Mosaic (600 images) (done)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.45

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex9_045.zip**)

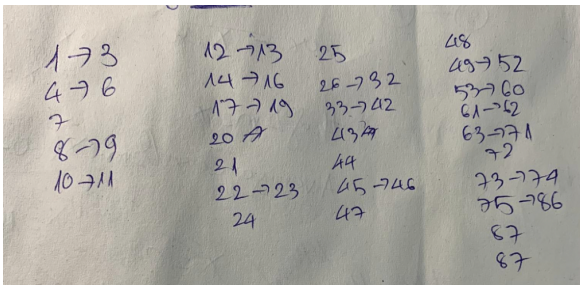
- The total number of bounding boxes after augmentation are too low (only near 4500) in experiment 9 compared to nearly 8500 in experiment 8

—> Separate only incorr & non_mask is not works —> Need to apply also for "mask"

▼ **Experiment 10 (2000 augment)**

- [x] ~~Shuffle between train & validation dataset & split the dataset into 7/3~~
  - [x] ~~Create a folder containing all images of both train & Val, 1 for labels also~~
  - [x] ~~Shuffle image's id & split them out to 7 for train & 3 for Val~~
- [x] ~~Try to apply augmentation on a new dataset~~

▼ Analyze the concepts in the "Public_test" folder



> "Train" & "Val" datasets are mixed together, shuffle & split out with a ratio of 7/3 (7 for train & 3 for Val)

▼ **Version 1 (not separate)**

Perform augmentation on folder contained images of both training & validation dataset

- ToGray + HorizontalFlip (200 normal images)

- GaussianBlur or medianBlur (400 normal images)
- Cutout (400 normal images)
- HueSaturationValue or RandomBrightnessContrast (400 images)
- Mosaic (600 images)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.43 (in progress)

—> Score on 40% of private set = mAP@.5 = 0.444 (name submit: **dataset_aug_ex10_v1**)

- **(NOTE)** This experiment violate the rules that "The training dataset interferes with the validation dataset" —> It makes the model perform well on the validation dataset after a few epoch

▼ **Version 2 (not separate)**

Perform augmentation on folder contained images of both training & validation dataset

The ***validation dataset is set for the public_test dataset***

- ToGray + HorizontalFlip (200 normal images)
- GaussianBlur or medianBlur (400 normal images + 100 images)
- Cutout (400 normal images)
- HueSaturationValue or RandomBrightnessContrast (400 images)
- Mosaic (600 images + 100 images)

▼ **Remark**

—> Score on Public set = mAP@.5 = (in progress)

—> Score on 40% of private set = mAP@.5 = 0.419 (name submit:)

- The result contains a lot of conflicts between "non_mask" and "incorrect_mask" when performing on public_test set
  - ☐ Recommend works: modify the bbox of incorrect_mask to better differentiate between non_mask & incorrect_mask

▼ **Version 3**

The same as version 2 but shuffle all the images in training dataset before training!!!!

▼ **Remark**

—> Score on Public set = mAP@.5 = (in progress)

—> Score on 40% of private set = mAP@.5 = (name submit:)

The reason why BTC limits to only 500 MB is b.c of the annotation!!!!

—> We can try harder on Cutmix to cut the box from data and paste them on another image —> To increase # of boxes!!!

(It's the key to winning the competition!!!) (Not effective 😓)

Week 5+6 (22/11 → 28/11, 29/11 → 05/12)

▼ **Really good resources!!!!!**

- All types of augmentations from basic concepts to advance concepts

https://www.kaggle.com/ishandutta/petfinder-data-augmentations-master-notebook

☑ ~~Separate images based on the number of labels in each image~~

- ☑ ~~Keep 50% of the dataset with the highest # of labels~~
- ☑ ~~The other images are augmentation~~

- [x] ~~Apply MixUp 50%/50% (Can not produce images)~~
- [x] ~~Shuffle the set of images before each type of augmentation (Shuffle is really important)~~

▼ **Experiment 11** (<3000 augmented images)

- [x] ~~Images having 2 or more labels (415 images) & images contained "incorrect_mask" (125 images) → merge them together (125+415 = 540 images) (Done)~~
- [x] ~~Validation dataset (actually public_test set) (88 images) (Done)~~
- [x] ~~100 images toGray & 100 images toGray + HorizontalFlip (200) (Done)~~
- [x] ~~Heavy augmented (600 images) (Done)~~

  HueSaturationValue OR RandomBrightnessContrast

  GaussianBlur OR MotionBlur

  CLAHE

- [x] ~~Cutout (350 images) (Done)~~
- [x] ~~Mosaic (Only from 2 labels or above) (600 images) (Done)~~
- [x] ~~Random BBox Safe crop (600 images) (Done)~~

▼ **REMARK**:

—> Score on Public set = mAP@.5 = 0.518 (best result) BUT only 0.433 (last result)

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex11_v1.zip**)

▼ **Experiment 12**

▼ **V2**

- Train_val folder used for train dataset
- New val folder (188 images)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.579 last & 0.603 best

—> Score on 40% of private set = mAP@.5 = (name submit: **.zip**)

▼ **V1**

- Train_val folder used for train dataset
- New val folder (188 images)
- 200 images toGray + horizontalFlip

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.413 last & 0.612 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex12_v1.zip**)

—> ToGray + HorizontalFlip only hurt model —> Use only 50 images

▼ **V3**

- Train_val folder used for train dataset
- New val folder (188 images)
- 50 images toGray + horizontalFlip
- 350 images Cutout

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.623 last & 0.659 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.48 (name submit: **submit_ex12_v3.zip**)

▼ **V4 (total 1908 images)**

- Train_val folder used for train dataset (882 images)
- New val folder (188 images)
- 50 images toGray + horizontalFlip
- 350 images Cutout
- Filter & add only incorrect-mask images once again
- 350 mosaic images

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.623 last & 0.659 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.501 (name submit: **submit_ex12_v4.zip**)

▼ **V5**

- Train_val folder used for train dataset
- New val folder (188 images)
- 50 images toGray + horizontalFlip
- 350 images Cutout
- Filter & add only incorrect-mask images once again * 2 = 75*2 = 150 images
- 350 mosaic images + 150 mosaic
  - Mosaic remove:
- 200 random BBox Safe crop
  - RBBSC remove: 5, 31, 54, 52, 65, 87, 104, 125, 128, 141, 158, 187, 189, 196, 208, 234, 271, 285, 299, 300, 332, 342, 375, 379, 382

▼ **Remark**

—> Score on Public set = mAP@.5 = last & 0.451 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex12_v5.zip**) → Not submit b.c problem with data

▼ **V6 (total 2320 images)**

- Train_val folder used for train dataset (882 images)
- New val folder (188 images)
- 50 images toGray + horizontalFlip
- 350 images Cutout
- Filter & add only incorrect-mask images once again * 2 = 75*2 = 150 images
- 500 mosaic (After fixed)
- 200 random BBox Safe crop (After fixed)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.448 last & 0.551 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.488 (name submit: **submit_ex12_v6.zip**)

- [x] ~~Filter only mosaic images having the clear shape of faces & exclude images having only a part of the face~~
- [x] ~~Do the same above with random BBox Safe crop (make 600 filters)~~

▼ **V7 (2950 images)**

- Train_val folder used for train dataset (900 images)
  - "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
- New Val folder (179 images)
  - The new validation set contains all images in public_test set + 5 images each group by using KNN to separate images in the training dataset into 20 groups —> 79 images + 100 images (KNN)

—> Set of images contained incorrect_mask label is in both "train" folder & "val" folder

- 50 images toGray (DONE)
- 350 images Cutout (DONE)
- Filter & add only incorrect-mask images once again * 2 = 88*2 = 175 images + horizontal (DONE)
- 502 mosaic (After fixed) (Done)
- 200 random BBox Safe crop (After fixed) (Done)
- 594 Rotation + Shear (prob 50/50) (Done)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.882 last & 0.888 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.528 (name submit: **submit_ex12_v7.zip**)

- "random BBox Safe crop" augmentation do NOT help

▼ **Experiment 13**

▼ **v1 (total 2678 images)**

- Train_val folder used for train dataset (997 images) (Done)
  - "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
  - Include also images having incorrect_mask labels in Kaggle dataset only
- New Val folder (179 images) (Done)
  - The new validation set contains all images in public_test set + 5 images each group by using KNN to separate images in the training dataset into 20 groups —> 79 images + 100 images (KNN)

—> Set of images contained incorrect_mask label is in both "train" folder & "Val" folder

- 50 images toGray (Done)
- 350 images Cutout (Done)
- Filter & add only incorrect-mask images once again (both original & Kaggle data) + horizontal() —> 200 images (Done)
- 502 mosaic (After fixed) (Done)
- 400 Rotation + Shear (prob 50/50) (Done)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.85 last & 0.88 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.477 (name submit: **submit_ex13_v1.zip**)

▼ **v2 (total 2858 images)**

- Train_val folder used for train dataset (997 images) (Done)

- "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
- Include also images having incorrect_mask labels in Kaggle dataset only

- New Val folder (179 images) (Done)
  - The new validation set contains all images in public_test set + 5 images each group by using KNN to separate images in the training dataset into 20 groups —> 79 images + 100 images (KNN)

  —> Set of images contained incorrect_mask label is in both "train" folder & "Val" folder
- 50 images toGray (Done)
- 350 images Cutout (Done)
- Filter & add only incorrect-mask images once again (both original & Kaggle data) + horizontal() —> 200 images (Done)
- 502 mosaic (After fixed) (Done)
- 400 Rotation + Shear (prob 50/50) (Done)
- 200 images shear + rotation with labels mask + non_mask only

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.865 last & 0.884 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.481 (name submit: **ex13_v1.zip**)

—> KAGGLE dataset not help in this competition!!!!!!!!!!!!!!

▼ **Experiment 14** (Not Kaggle dataset)

▼ **TODO**

- [x] ~~Combine training & validation folders into 1 folder~~
- [x] ~~Separate train & validation set based on KNN~~
- [x] ~~Delete all images of the public_test folder in the old val folder & add all remaining images into our new "train" folder~~
- [x] ~~If it lacks of face —> Don't work with that~~
- [x] ~~MAKE SURE all things we works since now are correctly labeled~~
  - [x] ~~Train folder (Done)~~
  - [x] ~~val folder (Done)~~
- [ ] Filter images with
  - [ ] With people behind the glass door!!!!!!
  - [ ] With images having only half face behind the door!!!
  - [ ] Filter them into a DataFrame so that use it to perform the augmentation

▼ **v1 (Total 2814 images)**

- Train_val folder used for train dataset (934 images)
  - "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
- New Val folder (154 images)
  - 66 KNN images + 88 public_test images = 154 images

  —> Set of images contained incorrect_mask label is in both "train" folder & "val" folder
- 100 images toGray (Done)
- 400 images Cutout (Done)

- Filter & add only incorrect-mask images once again * 2 = 90*2 = 175 images + horizontal (p=0.8) (Done)
- 367 mosaic + 84 mosaic = total 451 images (After fixed) (Done)
- 600 Rotation + Shear (prob 50/50) (Done)
  - 200 Rotation + Shear (prob 50/50) with non-mask only
  - 400 remaining images augmented normally

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.843 last & 0.843 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.497 (name submit: **submit_ex14_v1.zip**)

▼ **V2 (Total 2839 images)**

- Train_val folder used for train dataset (934 images)
  - "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
  - Finish relabel the dataset (99% correct label)
- New Val folder (154 images)
  - 66 KNN images + 88 public_test images = 154 images

—> Set of images contained incorrect_mask label is in both "train" folder & "val" folder

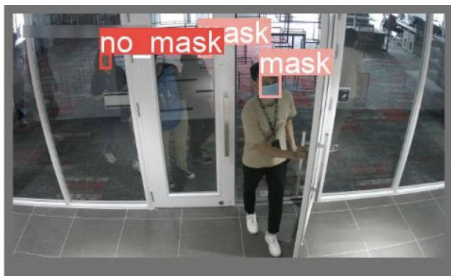
  - Finish relabel the dataset (99% correct label)
- 100 images toGray (Done)
- 400 images Cutout + HorizontalFlip(p=0.5) (Done)
- Filter & add only incorrect-mask images once again * 2 = 119*2 = 200 images + horizontal (p=0.7) (Done)
- 367 mosaic + 84 mosaic = total 451 images (After fixed) ()
- 600 Rotation + Shear (prob 50/50) (Done)
  - 200 Rotation + Shear (prob 50/50) with no-mask & mask only (Done)
  - 400 remaining images augmented normally (Done)

▼ **Remark**

—> Score on Public set = mAP@.5 = 0.881 last & 0.923 best (but in some initial epoch)

—> Score on 40% of private set = mAP@.5 = 0.533 (name submit: **submit_ex14_v2.zip**)

- Model fail to classify all the mages with people behind the glass door!!!!!!



- Also with images having only half face behind the door

▼ **v3 (Total 2939 images)**

- Train_val folder used for train dataset (934 images)
  - "train" folder contains all images having incorrect_mask labels from both original train_val data set and public_test set (!!!!!)
  - Finish relabel the dataset (99% correct label)
- New Val folder (154 images)
  - 66 KNN images + 88 public_test images = 154 images

  —> Set of images contained incorrect_mask label is in both "train" folder & "val" folder
  - Finish relabel the dataset (99% correct label)
- 100 images toGray (Done)
- 400 images Cutout + HorizontalFlip(p=0.5) (Done)
- Filter & add only incorrect-mask images once again * 2 = 119*2 = 200 images + horizontal (p=0.7) (Done)
- 367 mosaic + 84 mosaic = total 451 images (After fixed)
- 600 Rotation + Shear (prob 50/50) (Done)
  - 200 Rotation + Shear (prob 50/50) with no-mask & mask only (Done)
  - 400 remaining images augmented normally (Done)
- 100 images in below id list + horizontal()

  behide_door_id_lst =
  [102,20,13,27,57,66,160,192,218,217,219,408,674,801,632,923,222,428,560,545,874,881,229,509,531,575,584,576,87

▼ **Remark (V3_1) (HIGHEST SCORE)**

- Adding 100 images in behide_door_id_lst above + Horizontal() augmentation

—> Score on Public set = mAP@.5 = 0.912 last & 0.934 best epoch)

—> Score on 40% of private set = mAP@.5 = **0.575** (name submit: **submit_ex14_v3_1.zip**)

▼ **Remark (V3_2)**

- Take data from v3_2, add more 50 images horizontal() in the id's list above
- + 100 images with faces wearing different style masks

diff_mask_id = [12, 17, 225, 256, 39, 609, 659, 760, 797, 98, 160, 86, 634, 638, 639, 651, 650, 222, 223, 350, 661, 663, 678, 943, 518, 509, 531, 870, 871, 396, 313, 422, 626, 1037, 234, 323, 1038, 497, 586, 873, 621, 631, 644, 13, 662, 660]

- Remove 150 normally rotation() + Shear()

▼ **TODO**

- [x] ~~Delete all 400 shear + Rotation images~~
- [x] ~~Create again 200 shear + Rotation images with mask + no_mask label~~
- [x] ~~Create 150 shears + Rotation images normally~~
- [x] ~~Create 150 shears + Rotation images using behide_door_df~~
- [x] ~~Create 100 shears + Rotation images using diff_mask_df~~

- The result only nearly 0.88 lower than V3_1

—> There're 2 options here:

(1st) In the set of new list ids exit wrong labels one

(2nd) The model is general enough so that perform quite not good on Val folder but good on the whole private dataset

—> Score on Public set = mAP@.5 = 0.907 last & 0.943 best

—> Score on 40% of private set = mAP@.5 = 0.541 (name submit: **submit_ex14_v3_2.zip**)

▼ **Remark (V3_3)**

- Take data from v3_2, add 100 images horizontal() more in the id's list above
- + 50 images with faces wearing different style masks

▼ **TODO**

- [x] ~~Delete all 600 shears + Rotation images~~
- [x] ~~Create again 200 shear + Rotation images with mask + no_mask label~~
- [x] ~~Create 150 shears + Rotation images normally~~
- [x] ~~Create 200 shears + Rotation images using behide_door_df~~
- [x] ~~Create 50 shears + Rotation images using diff_mask_df~~

—> Score on Public set = mAP@.5 = 0.899 last & 0.902 best

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex14_v3_3.zip**)

▼ **Remark (V3_5)**

- Take data from v3_1
- + 150 images in below id list + horizontal()
- + 50 images with faces wearing different style masks
- Remove 200 normally rotation() + Shear()

▼ **TODO**

- [x] ~~Delete all 600 shears + Rotation images~~

- [x] ~~Create again 200 shear + Rotation images with mask + no_mask label~~
- [x] ~~Create 200 shears + Rotation images normally~~
- [x] ~~Create 150 shears + Rotation images using behide_door_df~~
- [x] ~~Create 50 shears + Rotation images using diff_mask_id~~

—> Score on Public set = mAP@.5 = 0.919 last & 0.948 best

—> Score on 40% of private set = mAP@.5 = (name submit: **submit_ex14_v3_5.zip**)

▼ **Main purpose!!!!!!!**

- [x] ~~Design the public_test & validation set which nearly successfully represents the private dataset!!!!!~~
  - [x] ~~It should contain all the concepts inside the training dataset~~
  - [x] ~~List 2 images for each concept in training dataset (With only the average # of labels only)~~

▼ **TODO**

- [x] ~~Shuffle all the files in the "train" folder before training (Important!!!)~~
- [x] ~~Construct the validation dataset based on KNN function code (Important!!!)~~
- [x] ~~Examine again the labels in the dataset (Maybe the bounding box should cover the ears also to differentiate between mask & incorrect_mask) (Important!!!)~~
- [x] ~~After relabeling, this is the final relabel version, we will design the validation dataset that contains all the concepts that appeared in the total dataset (!!!!!!! Final result)~~
  - [x] ~~Construct the Val dataset~~
- [x] ~~(**Interesting finding!!!**) Different types of image's shapes show different types of cameras used in the office —> We can group the images based on their image's shape~~

▼ **Experience when relabeling dataset**

- Need to confirm clearly the concept of each type of label (Mask, Non-mask, incorrect_mask)
- The model learned based on the pattern —> Need to guarantee each type of label has its own unique pattern (description)
- Only draw the box with enough place, More broad box → more info → Hard for the model to learn
- Next time when labeling, only dividing the labeling into smaller part (~ 100 images) is enough, then each person label 1 part.

▼ **Kaggle ideas about Data augmentation**

- Use the RandomBBoxSafeCrop augmentation technique to resize the images without hurting the bounding box

```
 A.Compose(
        [
            A.RandomSizedCrop(min_max_height=(800, 1024), height=1024,
width=1024, p=0.5),
            A.OneOf([
                A.HueSaturationValue(hue_shift_limit=0.2, sat_shift_limit=
0.2,
                                     val_shift_limit=0.2, p=0.9),
                A.RandomBrightnessContrast(brightness_limit=0.2,
                                           contrast_limit=0.2, p=0.9),
            ],p=0.9),
            A.ToGray(p=0.01),
            A.HorizontalFlip(p=0.5),
            A.VerticalFlip(p=0.5),
            A.RandomRotate90(p=0.5),
        A.Transpose(p=0.5),
        A.JpegCompression(quality_lower=85, quality_upper=95, p=0.2),
        A.OneOf([
        A.Blur(blur_limit=3, p=1.0),
        A.MedianBlur(blur_limit=3, p=1.0)
        ],p=0.1),
            A.Resize(height=1024, width=1024, p=1),
            A.Cutout(num_holes=8, max_h_size=64, max_w_size=64, fill_value=0,
p=0.5),
            ToTensorV2(p=1.0),
        ],
        p=1.0,
        bbox_params=A.BboxParams(
            format='pascal_voc',
            min_area=0,
            min_visibility=0,
            label_fields=['labels']
        )
    )
```

Github summary

- [x] ~~Utils~~
- [x] ~~Mosaic~~
  - [x] ~~Visualization~~
  - [x] ~~Mosaic construction & save notebook~~
- [x] ~~Data augmentation notebooks (visualization!!)~~
- [x] ~~Data visualization~~
- [x] ~~KNN notebook (2 types of using KNN) (In process)~~
  - [x] ~~Using features resulting from the VGG16 model (5 groups)~~
- [x] ~~Aug_experience (13)~~