

Brief Summary of data-preprocessing meet held on 12th March 7pm CET

- Both teams (keras and alumentations team) will work accordingly to create augmented images of each folder (that is CelebA, FFHQ, ATTGAN,...) separately.
- Few augmentation parameters to be followed-
 - Resize images to = (160, 160) (**must be same for both teams**)
 - Random rotation (0 to 360)
 - Horizontal or Vertical flip
 - Shear Range(0.2), zoom range(0.2) (can vary for both teams)
 - Height or width shift by 0.2 (can vary)
 - Blackout of either eyes, face or mouth (**very imp**)
 - Image compression (lower limit = 50, upper limit = 100)
 - Gaussian Blurring or Gaussian Noise (either of both or together also can be used, just make sure the image is not degraded or blurred a lot!)
 - **People working on Deepfake image folder, keep value of blurring and noise less as images are already blurred!** Just have a look at images properly before applying Blur or Noise
 - Either Random brightness, saturation, hue contrast (comes under ColorJitter in alumentations)
 - Avoid **cropping** or even if you do, use only central crop and that too with low probability. (to avoid information loss)
- **(Using too many augmentations can even degrade the dataset!)**
- Don't **Normalise/rescale** images right now. This will be done later while model training.
- Exploit features like OneOf provided in Alumentations library.
(link → [Composition API \(core.composition\) - Alumentations Documentation](#))
- For Alumentation Team :
 1. [Overview and visualization of pixel-level transforms from alumentations package | Data-science-blog](#)
(It covers all the features with examples.)
 2. [Index - Alumentations Documentation](#)
 3. **Very Important read** before setting values of probabilities and before using OneOf :
[Setting probabilities for transforms - Alumentations Documentation](#)
- For Keras Team:
 - Keras pre-processing layers
([Image preprocessing layers](#))
 - Keras Image Augmentation Layers
([Image augmentation layers](#))
 - More links -
[A Complete Guide to Data Augmentation | DataCamp](#)
[Data augmentation | TensorFlow Core](#)

General approach:

- Everyone is required to augment only 1 folder (at least in the Alumentations team!)
- **People augmenting Real images** (that is either 0-CelebA or 0-FFHQ folder) →
 - 5K real images need to be converted to 15K
 - For each image generate 3 augmented images
 - Save all augmented images in a separate folder, do not save any augmented images in google drive.
 - After augmenting, try to split into train, val, test (80%, 10%, 10% respectively). You can use “split-folder” ([split-folders · PyPI](#)) library if required.
 - Save this folder on your computer,
 - later on, upload it to the dropbox.
 - In case you are not able to split, upload the entire augmented folder onto dropbox and me or Akash will take care of it.
- **People Augmenting Deepfake images** (that is either 1-ATTGAN or 1-StyleGAN or 1-StyleGAN2 or 1-GDWCT or 1-StarGAN) →
 - 1K deepfake images need to be converted to 6K deep fakes
 - For each deepfake image generate 6 augmented images
 - Save all augmented images in a separate folder, do not save any augmented images in google drive.
 - After augmenting, try to split into train, val, test (80%, 10%, 10% respectively). You can use “split-folder” ([split-folders · PyPI](#)) library if required.
 - Save this folder on your computer,
 - later on, upload it to the dropbox.
 - In case you are not able to split, upload the entire augmented folder onto dropbox and me or Akash will take care of it.
- Write your own transformations, this way we will have variations in augmented images within each folder as well!

Link for google colab file -

<https://colab.research.google.com/drive/1dwzyaJAD2FG1XTHVKoU65PBRluz3n-fZ?usp=sharing>

- **Follow this procedure** →
(Preferably, work with openCV library)
 - while looping through each image, apply blackout on this image (let's call it blackout img)
 - From this blackout img, generate 3 real images(if you are augmenting either of real image folder) **OR** 6 deepfake images (if you are augmenting either of deepfake image folder)
 - You can use simple looping for above step
 - In each loop, save the image to the folder that **you** created (not to drive!).
Simply use `cv2.imwrite(...)` for this
- Now, split into train, val, test if possible.
- Upload on dropbox

