

Nom/Prénom: NEHME Ali
Master 2: Informatique ISE
Numéro étudiant: 15600238

Rapport: Corpus, fouille de données et apprentissage

Titre: Prédiction de célébrité à partir de tiny Yolo et de VGG-16.

Data : Les données utilisées pour ce projet proviennent de google Image. Deux sets de données ont été créés.

Donnée pour VGG-16 : Les mots clés que j'ai utilisés sont : « "Tom cruise, Brad Pitt, Jennifer lopez, Bill Gates, Donald Trump, Jean pierre coffe, Marine lepen, jacques chirac, jean lassalle ». On récupère alors, au mieux, 500 images pour chaque personnes. Cela fait, je nettoie ces images en détectant chaque têtes puis en positionnant chaque têtes droite (si la tête est penché) à partir de Dlib et je redimensionne la ROI avec une image de 224*224 (format de VGG-16).

Donnée pour tiny Yolo : Les mots clés utilisés ont été « Moroccan girl, american men french men, russian girl, algerian girl, german girl, Moroccan man, american women german men, russian men, arab girl, indian man, algerian men, arab men, indian girl Kpop man, french girl, kpop girl ». On récupère alors un set de 1000 images en tout. Pour créer les labels (ou annotations), j'ai utilisé « labelImg » afin de générer des fichiers xml en sélectionnant, pour chaque images, les ROI (Region Of Interest) à la main, c'est à dire où se trouve les visages dans l'image. La taille de l'image et la taille des ROIs n'importe peu.

Description du projet : Ce projet permet avant tout, de repérer un visage à partir de Yolo (tiny yolo a été utilisé) sur une image ou à partir d'une webcam. Ensuite, de prédire, à partir du visage trouvé, le type de « célébrité » que mon modèle a appris en utilisant VGG-16: « "Tom cruise, Brad Pitt, Jennifer lopez, Bill Gates, Donald Trump, Jean pierre coffe, Marine lepen, jacques chirac, jean lassalle, other^1, ali^2 ».

Inspiration:

Scraping:

<https://github.com/hardikvasa/google-images-download>
https://pypi.org/project/google_images_download/2.3.0/

Label cleaned:

<https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/>
<https://towardsdatascience.com/facial-mapping-landmarks-with-dlib-python-160abcf7d672>
<https://www.pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/>
<https://github.com/tzutalin/labelImg>

VGG-16:

<https://medium.com/datadriveninvestor/creating-art-through-a-convolutional-neural-network-ed8a4d9a3f87>

<https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5097666-tp-implementez-votre-premier-reseau-de-neurones-avec-keras>
<https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/>
https://moodle.univ-paris8.fr/moodle/pluginfile.php/403998/mod_resource/content/1/Lecon%204%20convNets%202019.pdf

Yolo:

https://medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899
https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
<https://medium.com/@venkatakrishna.jonnalagadda/object-detection-yolo-v1-v2-v3-c3d5eca2312a>
<https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/>
<https://github.com/thtrieu/darkflow>
<https://pjreddie.com/darknet/yolo/>

Other:

<https://towardsdatascience.com/step-by-step-guide-to-building-your-own-neural-network-from-scratch-df64b1c5ab6e>
<https://becominghuman.ai/classification-part-1-intro-to-logistic-regression-f6258791d309>
<https://medium.com/@curiously/tensorflow-for-hackers-part-iii-convolutional-neural-networks-c077618e590b>
<https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/>
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
<https://towardsdatascience.com/cifar-10-image-classification-in-tensorflow-5b501f7dc77c>
https://moodle.univ-paris8.fr/moodle/pluginfile.php/403998/mod_resource/content/1/Lecon%204%20convNets%202019.pdf
<https://www.pyimagesearch.com/2019/07/08/keras-imagedatagenerator-and-data-augmentation/>