Interpretation of neural networks and advanced image augmentation for visual control of drones in human proximity

Thesis Summary, Master in Computer Science, February 2021

Marco Ferri - m.ferri17@campus.unimib.it

***Abstract*—For this thesis, we consider the task of predicting the pose of a person moving in front of a drone, using the input images coming from an on-board camera. We aim to improve the generalization capabilities of a neural network [1] designed for this intent.**

First, we understand the main issues of the learned task through network interpretation. Then, we propose a solution to the generalization problem by applying data augmentation and background replacement to the original dataset. We retrain the model on the new data and compare the results of our solution with the original model's behavior.

Our entire work and additional details are available at: https://github.com/mferri17/cnn-drone-befree.

I. Introduction

In 2019, researchers at IDSIA have published a paper [1] which explores three methodologies for controlling a drone to closely interact with a person using neural networks models. This thesis focus on enhancing the model's generalization capabilities.

A. Task

The goal of [1] is to control a drone for continuously hovering in front of a moving person. The problem is approached as a reactive control procedure and addressed with supervised learning. In the paper, the authors propose three different approaches. For this thesis, we only focus on the mediated one, which aims to predict the pose of a person from an image.

The frames produced by an on-board drone's camera are used as input for a custom-designed ResNet [2] architecture. The model performs a regression on the four variables that compose the user's pose with respect to the drone (`X`, `Y`, `Z`, and yaw indicated with `W`). The network accepts a single 60×108 pixels image in input. To enhance readability, this model will be called *FrontalNet*.

B. Environment

The drone used by [1] and this thesis is a Parrot Bebop 2. It has a camera that can shoot 14 MP photos and record Full HD 1080p videos at 30 FPS. A Motion Capture (MoCap) system is used to track the drone and the user's movements in 3D space. At IDSIA, the drone arena's MoCap is composed of 12 infrared OptiTrack cameras for tracking passive markers' movements, placed both on the person's head facing the drone and on the drone itself.

C. Dataset

During data collection, the drone is controlled through an omniscient controller that knows both the drone's and the person's pose, given by the OptiTrack system. The controller script computes acceleration commands for the drone for making it hovering in front of the person, facing the user's head orientation at a predefined 1.5 meters distance.

Data collected inside the arena are used for training of a machine learning model. The dataset counts about 63'000 and 11'000 frames for training and testing sets, respectively. The ground truth is represented by the four coordinates `X`, `Y`, `Z`, and yaw (indicated with `W`), associated with each captured image.

D. Baseline Performance

FrontalNet is trained using the Mean Absolute Error (MAE) loss function with the ADAM optimizer and a base learning rate of 0.001. An official test set is used to evaluate the model through the coefficient of determination (R^2) metric. According to the paper, predictions seem more accurate for variables `Z` and `W` with an R^2 score of 0.82 and 0.88, respectively. Quite different the results for `X` and `Y` which only respectively reach an R^2 of 0.59 and 0.57.

Moreover, [1] reports qualitative experiments conducted inside the drone arena by flying the drone without the MoCap system, hence only relying on the learned model for computing the user's pose. The outcome is excellent, with the drone actually performing its task without any issues.

E. Generalization Issues

For both quantitative and qualitative evaluations, [1] uses a set of images that is similar to the training one. For a complete analysis, we must consider model performance in unknown environments. The paper does not address the topic, but during a direct discussion with the author, we discovered that flying performance outside of the drone arena was not consistent with the usual model behavior. The drone was not able to follow the user appropriately, and its movements were unpredictable. We conclude that the network is not able to accomplish its task outside of the training environment.

F. Frameworks

The original work from [1] is written in Python 3 and based on ROS, TensorFlow 1, and Keras. We adapt the code for working with TensorFlow 2. Here is a list of the main libraries used:

- `Numpy`, for computation on arrays;
- `Pickle`, for saving and loading Numpy arrays;
- `Matplotlib`, for building charts and visualize images;
- `OpenCV`, for efficient image/video manipulation;
- `TensorFlow 2`, crucial for the entire project: network interpretation, person masking, training, and evaluation;
- `Keras`, for defining the network architecture, training, and evaluating the model;
- `TensorBoard`, for profiling training time;
- `Sklearn`, for computing some evaluation metrics;
- `Albumentation`, for implementing image augmentation,
- `tf-keras-vis` [3], for applying GradCAM and other interpretability techniques;
- `akTwelve Mask_RCNN` [4], for human detection and segmentation during background replacement.

II. Solution Design & Implementation

A. Model Interpretation

Convolutional Neural Networks (CNN) are said "black-boxes" because their reasoning and comprehension are intrinsic in the network parameters, which are nothing but numbers, particularly hard to understand even for domain experts. Explainable Artificial Intelligence is the field of study which tries to make machine learning results and their underlying basis for decision-making properly interpretable to humans.

Among interpretability techniques, we select Grad-CAM [5], whose application on a specific image produces a heatmap. Overlapping the heatmap with the input image, the algorithm enables the visualization of the parts of the image which are actually responsible for predicting a certain model's output. For this reason, Grad-CAM is indeed the most understandable way of visualizing what a CNN is learning. The algorithm is designed to be applied to classification tasks rather than regression ones. Thus, for network interpretation only, we decide to transform our problem into a classification task[1].

When analyzing Grad-CAM results, we want the heatmaps to mainly overlay the person in the images. If so, we can conclude that the network effectively understands the concept of a person and produces its outputs based on the user's position only. Otherwise, we realize that the predictions are driven by undesired factors.

Figure 1 reports some examples of correctly made detections, in which only the person in the image is identified. Such precise results are not the standard. In many cases, the network focus is unstable, and the heatmaps frequently go in and out of the target person. Furthermore, our network interpretation also reveals many flaws in the prediction task. Grad-CAM exhibits several situations in which the model output is affected by recurrent elements in the dataset. Figure 2 clearly shows how the FrontalNet model is attracted by many elements in the background.

We conclude that the model is not robust enough to only focus on the user who is actually facing the drone's camera. Instead, various portions of the input images are considered when the model makes its predictions. We can reasonably assume that the ResNet has undesirably learned some details about the drone arena in which the dataset has been collected.

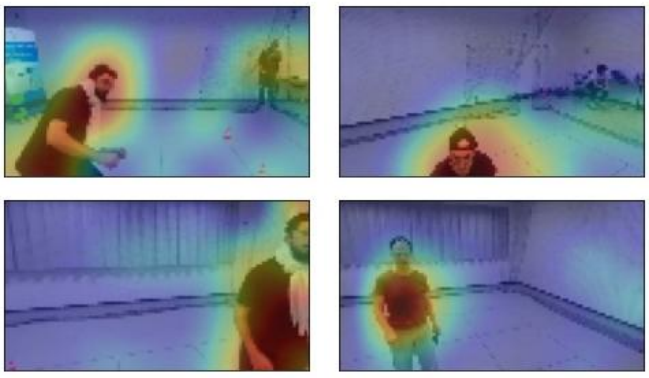

Fig. 1. Grad-CAM: Correctly detected people.

[1]technical details on the transformation from regression to a classification task are omitted for shortness, but still available in the full thesis

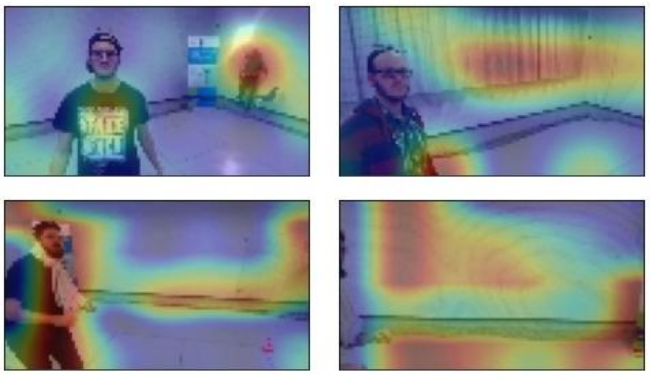

Fig. 2. Grad-CAM: The model gets easily distracted by various elements in the background: people, curtains, baseboards, or even walls spots.

B. Background Replacement

We have seen that the principal cause of the problem is inherent in the drone arena; thus, we would like to remove this factor (i.e., the room itself) from the equation. We propose a solution, inspired by Domain Randomization [6], that consists of performing advanced data augmentation on the training data. We decide to keep only the users and replace the backgrounds in the images.

After several experiments, the solution we have decided to adopt is Mask R-CNN [7], a state of the art deep learning framework for object detection and instance segmentation. The algorithm detects and creates a mask for all the objects appearing in the input images, labeling each mask with the category to which the object belongs (e.g., person, TV, bike, car, etc.). Results on our dataset are incredibly precise, and the method undoubtedly outperforms any other previously experimented since it provides both human detection and segmentation at once. Figure 3 presents how Mask R-CNN easily detects people in our video frames, regardless of their low resolution. This high-level accuracy comes with an extremely-high computing power requirement[2], which requires the inference to be made offline.

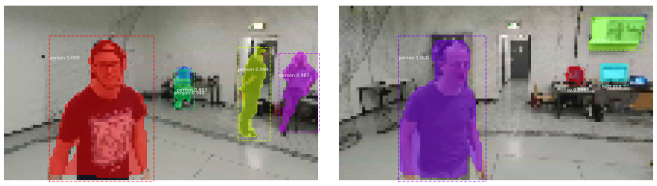

Fig. 3. Mask R-CNN applied to our training set.

To perform the background replacement, the training procedure receives, with each sample, also the corresponding user's mask. This is used to distinguish the subject from the rest of the image and accordingly blend the camera's frame with another image, serving as the background. The ground truth remains unchanged, and this is the main advantage of our approach. We can simulate a dataset acquired in different environments without actually collecting it, which would otherwise require a dedicated MoCap system.

By providing a considerable amount of images to use as backgrounds, the machine learning model trained on the modified dataset

[2]according to [7], Mask R-CNN runs at 5 FPS on Nvidia Tesla M40 GPU

should be able to actually ignore the background and, instead, hopefully learn the concept of a person.

We select the dataset for Indoor Scene Recognition [8] presented during the 9th Conference on Computer Vision and Pattern Recognition (CVPR). The dataset contains a total of 15'620 images divided into 67 indoor categories. For shortness, the dataset will be referred to as *CVPR*. During training, each sample is assigned to a randomly chosen background from the CVPR dataset. Even though the background replacement is done on the fly, the combination of each input image with a background forms a brand new dataset. Figure 4 shows a brief demonstration of the background replacement technique applied to some training images.

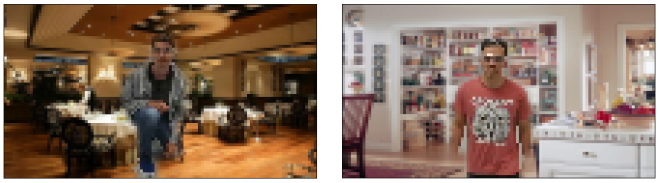

Fig. 4. Example of background replacement on the training set.

C. Data Augmentation

Image augmentation is ubiquitously used with CNNs for reducing overfitting on the training images by applying random transformations. For the implementation, we choose Albumentations. Being not able to modify the ground truth with affine transformations because the relationship between the person's position in the image and the ground is not known a priori, spatial-level augmentations are not an option. Instead, we mainly apply pixel-level transformations. The only exception is horizontal flipping, which only requires inverting the `Y` coordinate in the ground truth.

The combination of different Albumentations transformations composes a pipeline, whose time performance depends on custom choices and probabilities. The designed pipeline takes about 4 seconds to process 10'000 images and accepts a parameter to define the prior probability of applying the augmentations to an input image. Furthermore, at the end of the pipeline, we also apply Perlin noise.

D. Models Definition

We are interested in the beneficial effects on generalization caused by background replacement, which can be amplified when it is combined with image augmentation. We define three model alternatives to evaluate the validity of our solution.

- A baseline model trained on the original dataset as-is, corresponding to the exact approach proposed by [1]. Being trained with data coming the drone arena, we call it the `Arena` model.
- A first variant of the above model, defined by enabling only background replacement. In this case, we replace original backgrounds with randomly chosen images. We use the CVPR dataset, therefore we call this model `CVPR`.
- A second variant of the model includes both background replacement and image augmentation. This combination constitutes the most advanced model proposed in our thesis, called `CVPR Aug`.

E. Training

Through the `tf.data` API provided by TensorFlow, which allows the definition of complex and modular input pipelines, we develop a custom data generator responsible for retrieving and pre-processing the dataset during training. The library offers the possibility of optimizing the operations through asynchronous execution on the GPU, reducing the total idle time. Optimizations are crucial when working with complex input pipelines, such as for the `CVPR` and the `CVPR Aug` models. We use TensorBoard to profile the training procedure and evaluate the actual improvements provided by our choices. We discover that the non-optimized versions of the generator are highly input-bound (up to 47%), against a 0.2% of average time spent waiting for the input when optimizations are enabled.

As in [1], we use the MAE loss function and the ADAM optimizer with the default learning rate of 0.001 and automatic reducer on plateaus. The batch size is 64, and the last 30% of the training data (not shuffled) is used as a validation set. The training runs for 60 epochs, without early stopping. The training set is shuffled and repeated three times for each epoch for imitating an oversampling strategy, especially suited for the `CVPR` and the `CVPR Aug` models. The latter is trained with a prior augmentation probability of 95%.

We train the models on a dedicated workstation available at IDSIA[3]. We observe the following average time performance.

- `Arena` model takes 20 minutes (70% GPU utilization).
- `CVPR` model takes 40 minutes (50% GPU utilization).
- `CVPR Aug` model takes 120 minutes (20% GPU utilization).

All the models require 1724MiB of constant memory usage. Training time is inversely proportional to GPU utilization since more data pre-processing operations, usually performed on CPU, also require a greater amount of time. Background replacement is implemented with TensorFlow and executed on GPU, thus only doubles the training time (`CVPR`). On the contrary, Albumentations has to be run on CPU, significantly increasing the total time required (`CVPR Aug`).

III. Evaluation

A. Training Results

This section considers the models' performance during training. The MAE loss function and the R^2 score are taken into consideration. At least 30 epochs are needed to reach convergence, after which the performance stabilizes. For shortness, only the charts related to the `CVPR Aug` model are shown in figure 5.

`Arena`, `CVPR`, and `CVPR Aug` respectively achieve a training/validation loss of 0.03/0.07, 0.08/0.13, 0.20/0.20. As the model complexity increases, the loss does. The `CVPR` and the `CVPR Aug` models reach a validation R^2 of 0.96 and 0.92, respectively. However, the `Arena` model registers an R^2 of 0.99, which can be easily attributable to overfitting.

B. Quantitative Testing

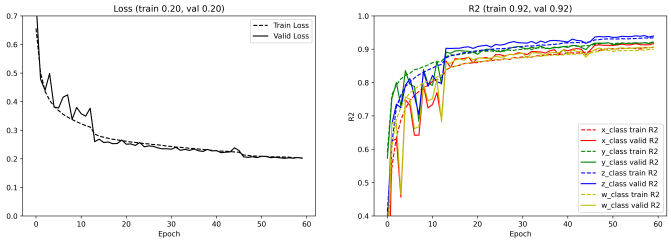

Fig. 5. `CVPR Aug` model's MAE and R^2 during training.

[3]Workstation specifications: OS Ubuntu 18.04 64 bit; CPU Intel Xeon Gold 5217 octa-core @ 3 GHz; GPU Nvidia GeForce RTX 2080 Ti (11 GB of dedicated memory); RAM 128 GB 2666MHz.

For testing, we expand the official test set [1] through background replacement, selecting two indoor scenarios not present in the augmented training set. The respective datasets, created by replacing all the test images backgrounds, are called `indoor1` and `indoor2`; we call `arena` the original test set. We test our three model variants on the three test sets. Unlike for training, during testing, we evaluate the R^2 score separately on each variable, allowing us to inspect models' behavior further. Besides, we compute Root Mean Squared Error (RSME), particularly useful for understanding the errors' magnitude. Results obtained by each model are summarized below.

1) Arena model: Baseline performance on the test set. On its native dataset (i.e., the `arena` test set) the loss is 0.41 and the R^2 lies between 0.74 and 0.86. As expected, the model behavior on background-replaced test sets is very poor. On the `indoor1` dataset, the loss is 0.86 and the R^2 scores ranges between 0.08 and 0.30. Even worse, on `indoor2` the loss is 1 and R^2 registers negative values for variables `X` and `Z`.

2) CVPR model: The model performance on `arena` test set are very similar to the one obtained on it by the `Arena` model. The only difference is found for the `X` R^2, which decreases from 0.81 to 0.69. Howbeit, `CVPR` is better than `Arena` when predicting the user's pose with unknown backgrounds. In fact, `indoor1` and `indoor2` test sets registers a loss of 0.44 and 0.45, respectively.

3) CVPR Aug model: Image augmentation seems to provide not only general improvements but also leverages the outcome for all variables. The `CVPR Aug` model achieves an R^2 score always greater than 0.80. Also, its performance on the `arena` test set is the best so far, with a loss of 0.36. Relative R^2 scores go from 0.75 to 0.87, with no particular differences on different test sets.

Notes on variables: Overall, all the models report consistent trends with respect to their variables. In all cases, the RSME associated with `W` is considerably higher than the others. Accordingly, the respective R^2 is lower. Considering the R^2, `Y` and `Z` are the best predicted variables, followed by `X`. Also, `X` receives a huge advantage from the image augmentation (`CVPR Aug`).

Final considerations on quantitative evaluation: Our strategy undoubtedly improves the original model from a numerical perspective. The approach seems able to uncouple the training images from the generalized task of recognizing the user's position. Data augmentation plays a fundamental role in enhancing the robustness of the model. `CVPR Aug` model even registers better metrics than the original model inside the drone arena, probably because such an environment is relatively more comfortable to handle than the CVPR backgrounds used during training.

C. Qualitative Testing

A first evaluation presents a comparison with timeline charts. They investigate the general trend of each model on different test sets. Models `Arena`, `CVPR`, and `CVPR Aug` are shown in green, blue, and red, respectively. Through background replacement, we create a new test set composed of multiple backgrounds rather than a single one. Thus, we call it the `mixed` test set. Consecutive images in the dataset are randomly assigned to one of the available backgrounds.

1) Test set `arena`: All the models achieve good results on the original test set. As in quantitative evaluation, there is no significant difference between the three models.

2) Test sets `indoor1` and `indoor2`: With artificial test sets, things start to change, as shown in figure 6. `CVPR` and `CVPR Aug` are both able to produce predictions very similar to the ground truth for all the variables. On the contrary, the `Arena` model (in green) only follows the ground truth during a tiny percentage of the time, and only for the variables `Y` and `W` (2nd and 4th timelines).

3) Test set `mixed`: Evaluation on the `mixed` set further provides evidence that the `Arena` model is computing predictions by considering the images' background. Its output continuously goes up and down arbitrarily, as shown in figure 7. This is due to the fact that the `mixed` test set has different backgrounds for subsequent samples. Thus, the oscillations are a symptom of the model sensitivity to the background, as demonstrated through Grad-CAM in section II-A. As before, `CVPR` and `CVPR Aug` models perform in the task as they do on the original test set.

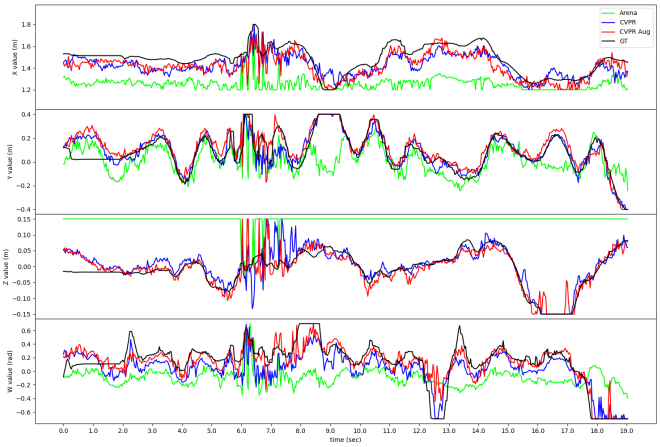

Fig. 6. Qualitative evaluation: GT vs. predictions, `indoor2` test set.

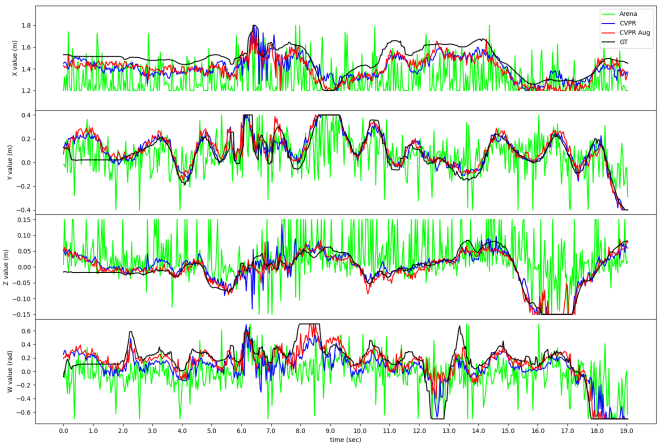

Fig. 7. Qualitative evaluation: GT vs. predictions, `mixed` test set. `Arena` model predictions are rapidly changing due to a high sensitivity on the images' background.

Additional evaluation

To complete the evaluation, we test the model on new images acquired through a smartphone. Having no available ground truth, the following assessment is only based on human interpretation. We rely on a visualization that simultaneously shows the considered frame and all the necessary information about predictions. The camera image is in the center, surrounded by the predicted values. Models `Arena`, `CVPR`, and `CVPR Aug` are shown with markers in green, blue, and red, respectively. For the analysis, we evaluate the markers' position over the variables axis to check whether they comply with the image. A summary of the variables' interpretation is shown below.

- `X`, on the left, is the distance from the camera. When the user is 1.5 meters distant, the marker is in the center. If the person is further, the marker goes up; otherwise, it goes down.
- `Y`, on the bottom, represents the horizontal alignment. Usually, the relative marker exactly maps the user's position in the image over the x-axis. In other words, when the prediction is correct, the `Y` marker is precisely under the person in the frame.
- `Z`, on the right, represents the vertical alignment. When the user's face is vertically aligned with the camera, the marker is centered. If the user lowers, the mark goes down; otherwise, it goes up.
- `W`, on the top, is the orientation of the head. When the user is perfectly facing the camera, the marker will be centered; otherwise, it will follow the user's sight. For example, if the person turns the head on his right, the marker will go left, following the user's eye.

We test the models' behavior in various conditions: in many scenarios (both outdoor and indoor), with different users, in several light conditions, and even with multiple people in the frame or with their face partially covered. The `CVPR Aug` model (in red) overall achieves the best results in tracking the user's pose in the image. Variables `X`, `Y` and `Z` generally follow the user properly, without many estimation errors. The `W` is more prone to fail on its task.

The main differences between the three models are observed for the `X` and the `Z` variables. This is especially visible when considering tough situations, such as people jumping or running (figure 8). The last case also demonstrates that the `CVPR Aug` model can surprisingly work with a person who is not actually facing the camera.

Some experiments also consider multiple people in the camera's frame (figure 9). In such situations, `CVPR Aug` sometimes correctly tracks the foreground person while other times does not explicitly choose one to follow.

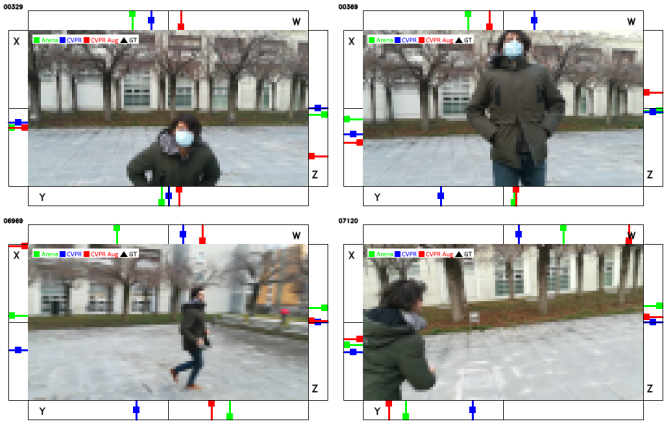

Fig. 8. Qualitative evaluation: models' behavior on jumping or running people. All the variables are correctly predicted in these examples, even when the person is not actually facing the camera.

IV. Conclusion

In this thesis, we presented a methodology for generalizing the ResNet defined in [1]. Designed to predict a user's pose from an image, the model was not able to achieve its task out of the training environment. We first applied Grad-CAM to understand what the model was actually considering while computing its prediction, and we discovered that the network was suffering from many biases coming from the training data.

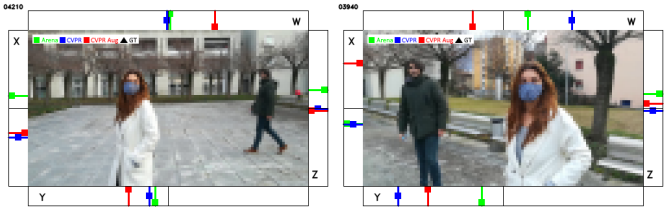

Fig. 9. Qualitative evaluation: models' behavior on multiple people. Variables `X` and `Y` are the most suited to be considered in such situations, as easily understandable by humans. In the first image, the girl's pose in the foreground is predicted by the model; in the second, the man in the background is considered. This is probably due to people distance, clothes and face mask.

As a solution, we proposed a modification of the dataset to reduce overfitting. Our approach consists of applying background replacement and image augmentation to the original images for the creation of new data. Retraining the original ResNet on the data obtained after our augmentation pipeline, we created an enhanced version of the model.

Our quantitative and qualitative evaluation demonstrated that our improved model is capable of solving the task in unseen environments. This is a demonstration that our approach improved the generalization capabilities of the original model.

A. Future works

Considered the results obtained during our evaluation, we can define a couple of future milestones:

- A test on the real drone is needed to further confirm our approach's success in the task. We want to ensure that our approach is effectively able to predict a user's pose in any environment through the real drone's camera. Experiments in various scenarios have the possibility of ultimately assess the model's generalization capabilities.
- A recent work in IDSIA [9] have brought the same task of human pose prediction on-board the Crazyflie, an ultra-low-power nano-drone. It would be interesting to apply our augmentation pipeline on that work to prove the power of the solution on different platforms designed to achieve the same task.

References

[1] Dario Mantegazza, Jérôme Guzzi, Luca Maria Gambardella, Alessandro Giusti: *Vision-based Control of a Quadrotor in User Proximity: Mediated vs End-to-End Learning Approaches*. (2019)

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: *Deep Residual Learning for Image Recognition*. (2015)

[3] Yasuhiro Kubota: *Visualization toolkit for debugging tf.keras models in TensorFlow 2*. https://github.com/keisen/tf-keras-vis

[4] Adam Kelly: *Mask R-CNN in TensorFlow 2* https://github.com/akTwelve/Mask_RCNN

[5] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, Dhruv Batra: *Visual Explanations from Deep Networks via Gradient-based Localization*. (2016)

[6] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J. Pal, Liam Paull: *Active Domain Randomization*. (2019)

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross B. Girshick: *Mask R-CNN*. (2017)

[8] A. Quattoni, and A.Torralba: *Recognizing Indoor Scenes*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009)

[9] Nicky Zimmerman: *Embedded Implementation of Reactive End-to-End Visual Controller for Nano-Drones*. (2020) — Master Thesis