Pedestrian Semaphore Classification An Active Learning Approach

Gabriel Fernandes[1], Miguel Rabuge[2], and Pedro Rodrigues[3]

[1,2,3]Department of Informatics Engineering, University of Coimbra, Portugal
[1]gabrielf@student.dei.uc.pt, [2]rabuge@student.dei.uc.pt, [3]pedror@student.dei.uc.pt

***Abstract*—The environments, in which pedestrians with some form of disability circulate are often extremely risky, more so than for "healthy" pedestrians. Not only the environments are dynamic, and full of uncertainty but, since an impaired individual is not able to record quality sensory information his ability to navigate through the environment, without endangering himself, is compromised. The most common solution for this problem is some form of personal companion, e.g "a guide dog", that helps the individual circulate. Training these kinds of companions is often slow and expensive... We propose an "Active Learning" (AL) approach for training a classifier that could be integrated in an artificial personal companion module. In our work, we go on to compare our AL methodology not only with state of the art models, but also with a model reported in recent paper on this topic. The obtained results show that models trained using this methodology can obtain the similar accuracies and with less data required.**

***Index Terms*—Artificial Intelligence, Machine Learning, Active Learning, Autonomous Pedestrian Module, Image Classification.**

I. Introduction

Nowadays, there is a lot of effort being put into solving the problem of autonomous driving, not only because of research but also because of the security and comfort that autonomous vehicles may bring to our lives. Despite the fact of this problem being hard, it is made easy because of the environment where the action of driving happens. There is a set of rules about how to drive, and where to drive to avoid potential accidents resulting in collisions with other vehicles, persons or objects. In other words, the environment is dynamic and risky, but it is less chaotic and more well-defined than others, which becomes more suited for the development of artificial intelligence approaches for training vehicles to be autonomous.

To our knowledge, there has not been a lot of development of similar techniques for pedestrians, namely the ones that show some form of disability that could put them in danger if not accompanied by some form of personal companion. The pedestrian environment, unlike the one where vehicles circulate, does not have a set of rules describing the way people should walk nor a place where the action of walking takes place. The pedestrian may be forced to walk on a sidewalk, a (lighted) crosswalk, in rugged terrain or even in the car lane if no sidewalk exists. Additionally, the objects that a pedestrian needs to avoid obviously increases since a person needs not to worry only about walking itself, but also other obstacles with more unpredictable moving patterns. Therefore, the environment is often so chaotic that travelling by foot requires assistance and makes the problem less approachable from an artificial intelligence standpoint.

Since the problem of autonomous mobility for disabled pedestrians is not particularly developed, and the only way such mobility can be achieved is with the help of some form of companion, (like for instance a guide dog which requires a lot of training to successfully guide people), our contribution is to advance inside this field in terms of pedestrian semaphore detection and classification for a crosswalk-taking decision. The choice of an active learning (AL) approach is justified by the lack of datasets for this specific task. Therefore, AL is the perfect methodology to apply for this niche, but yet significant, application, since it can extract better accuracy from the models with fewer data.

In short, with this project we aim to explore an alternative methodology for this problem, by comparing the State-of-the-Art methods performance with our own, using various metrics. Our main objective is to build a classifier that should be fast to classify and highly accurate, without the need for large amounts of data.

We will base our research of a late breaking work entitled - "Flying Guide Dog: Walkable Drones and Transformer-based Semantic Segmentation" [Tan+21a] - that uses a custom-made classification model based on a convolutional neural network (CNN) for semaphore detection on segmented images. Moreover, not only will we test our AL approach against this paper's CNN model, but we will build and train other State-of-the-Art CNN's in order to further study and justify the possibility of an AL approach to yield more efficient results.

II. State Of The Art

In the past few years, the State of the Art of image classifiers tends to revolve around Convolutional Neural Networks, due to the large amounts of data and the processing power currently available, which have turned out to be highly rewarding when both conditions are true. Although the hardware available today is really sophisticated and allows us to achieve good results through the use of CNN's for image recognition.

Regarding Active Learning, this approach aims to iteratively find the most informative data points to train the model, through the use of impurity measures. This methodology reveals being highly rewarding, generating good models with low data usage.

In the next sections we will discuss the state of the art convolutional neural networks, focusing our analysis on the "LeNet-5" [Lec+98] and "AlexNet" [KSH12] deep neural networks that we use later in work as a base for comparison and benchmark. We also refer the Active Learning methodology and point out some known strategies used with this method that we will also take advantage of in our experimental phase.

A. *Convolutional Neural Networks (CNN's)*

For the last years, deep learning has become a hot topic in the Artificial Intelligence landscape, with convolutional neural networks being one of the most widely used methods for image classification, natural language processing and computer vision tasks. Prior to this other methods were used that relied on slow and time-consuming feature extraction methods, that with CNN's are not necessary since they are able to work with raw image data without any need for feature pre-processing, therefore being perfect at picking up small patterns, which then are "processed" and combined through the multiple layers of the network, until the model can make sense of the provided data and classify what a given object really is.

Convolutional Neural Networks pose a great advantage in image classification, relying mainly on mathematical and algebraic models. In fact, many of the operations performed inside the multiple layers of these deep networks often involve matrix multiplications and other calculations that are highly vectorizable and parallelizable. The main drawbacks are that they are computationally demanding, often requiring graphical processing units (GPUs) for training more complex models and usually take some time to be trained to a level that make them efficient and accurate. In brief, by using CNN's the time and effort shifts from the user to the machine which somewhat advantageous, with the processing power we have available to us nowadays.

The architecture of a CNN is mainly based on a multi-layer feed forward network comprised of layers that learn from features processed by the layers before. The layers of these networks belong to one of these three types: convolutional, fully connected and pooling, existing many possible architectures made from arranging these in meaningful way.

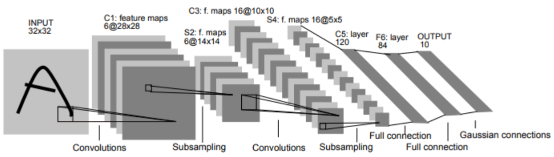


Figure 1. Example architecture of a convolutional neural network (LeNet-5); image from [Lec+98]

One well known attempt on CNN design was done by **Yann LeCun** with the self-titled LeNet-5 [Lec+98], which is known to have good performance in handwritten digit recognition (MNIST) dataset and is still to this day considered a reference. Overall this CNN consists on a five layer architecture that at a very high level is composed of two convolutional layers that serve as an encoder, and three fully connected layers that belong to the dense block of the network. A more recent attempt on a CNN design developed by **Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton.**, the "AlexNet" [KSH12]. The CNN consists on an implementation on the LeNet-5 [Lec+98] but much deeper, having a total number of layers that amount to eight. Architecturally, the first five layers consist of convolutional layers (oftentimes appearing interleaved with pooling layers) and the last three are fully connected layers.

These CNN designs have proven to have good performance and have been considered state of the art although other designs may be more effective for different datasets. For instance, on the "Pedestrian and Vehicle Traffic Lights" (PVTL) dataset [Tan+21b] the authors used their own implementation of a simple CNN (5 convolutional layers + 3 fully connected layers), claiming that it was able to achieve 83% accuracy; furthermore, by using a fine tuned CNN based on the "ResNet", another state of the art CNN provided in the *keras* python library, they were able to obtain accuracies of 90%, both over a 25 epochs training.

Overall the results obtained by using state of the art CNN based models for image classification are promising and, when properly tuned, tend to be even better suited for each specific need. Our aim, with our work, is to further explore other possibilities for improving these models through the use of other techniques, such as active learning.

B. *Active Learning*

On active learning **Burr Settles** [Set09; Set11] describes it as being a method for further improving the performance of machine learning models, making them achieve greater accuracies. In this semi-supervised learning method, through the usage of queries of unlabeled data made to an *oracle*, the active learner is not only able to choose the data from which it learns but also choose the data where the model is most uncertain about and learn from it, therefore increasing the learning process speed.

The active learning cycle is a sequence of steps by which an active learner attempts to iteratively improve the performance of the model. One of the most common active leaning cycle methodologies is the pool-based sampling active learning, in which the queries made to the *oracle* select examples from a "pool" of unlabeled instances in accordance to a given strategy that should take into account the information gain that a given instance may add to the model.

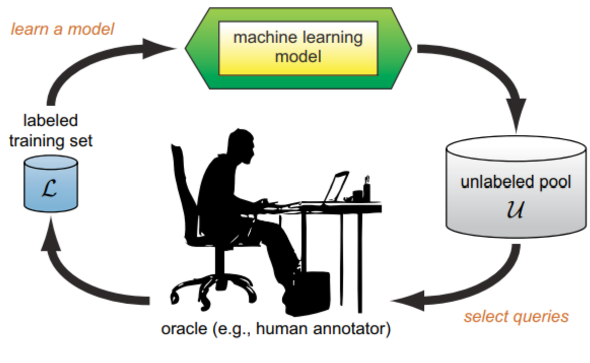


Figure 2. A pool-based active learning cycle; image from [Set09]

There exists a wide variety of query strategies, namely: query-by-committee, expected model change, expected error reduction, uncertainty sampling, etc... .One of the most commonly used query strategies, and that we will use in the experimental phase of our work is *uncertainty sampling* and consists in a framework where the queries to the model, made by the active learner, are done in such a way as to select the instances from the pool which the learner is *less confident* about (1). The standard measure of uncertainty used by this method consists in the choice of the instance whose prediction conveys the lowest confidence. Other measures often considered are the *entropy* (2) and the probability difference between the two most likely labels (3); these strategies correspond to the so called *entropy sampling* and *margin sampling* strategies respectively.

$$x^*_{LC} = \underset{x}{\operatorname{argmax}}\, 1 - P_\theta(\hat{y}_1|x) \tag{1}$$

$$x^*_{H} = \underset{x}{\operatorname{argmax}} \sum_i P_\theta(y_i|x) \cdot \log P_\theta(y_i|x) \tag{2}$$

$$x^*_{M} = \underset{x}{\operatorname{argmin}}\, P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x) \tag{3}$$

Above are shown in mathematical notation the possible metrics for picking an instance under the *uncertainty sampling* active learning framework. In the formulas above, θ represents the model and x, y instance labels. For formulas (1, 3), $\hat{y}_k = \operatorname{argmax}_y P_\theta(y|x)$ denotes the k-th most probable instance label under the model θ. [Set09].

Having discussed the state of the art active learning methodologies we will now move on to the experimental phase of our work where we will put these methods in practice and check if we can extract any benefits of their usage.

III. Materials: Datasets and Frameworks

In terms of pedestrian assistive agents, there has not been a lot of research related to it since a "Pedestrian Assistant" is a relatively recent topic. The most recent scientific paper that we found for it, uses a standard state of the art CNN for the pedestrian traffic lights detection and classification module of their project, trained by using a newly introduced dataset "Pedestrian and Vehicle Traffic Lights" (PVTL), [Tan+21b] which is a preprocessed subset of Mapillary's vast image data set that we are going to make use throughout our work.

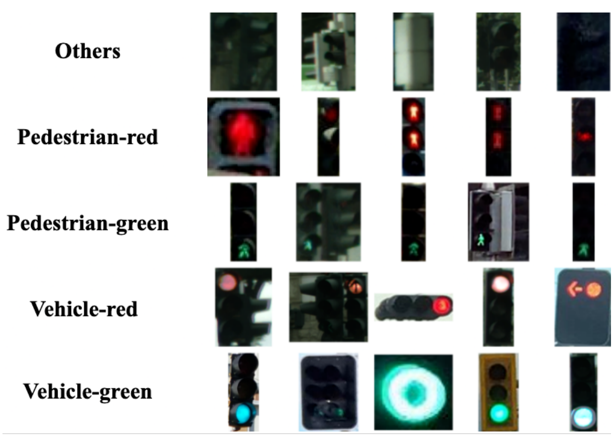


Figure 3. Pedestrian Vehicle and Traffic Lights dataset sample; image from [Tan+21b]

This dataset is composed by a series of images of semaphores of different types, colors and shapes. The data in this dataset is grouped into five different classes: "Others", "Pedestrian-green", "Pedestrian-red", "Vehicle-red", "Vehicle-green". The image above (figure 3) illustrates the different dataset classes and shows some samples of images that may be found in each class. Each one of these classes contains about 300 images that we preprocessed in order to standardize the input. The image preprocessing step will be discussed in the "Dataset Preprocessing" section III-A.

The whole project was developed using the python programming language and making use of multiple python frameworks that provided us with tools for image processing, deep neural network model construction/training and active learning. A list of frameworks used in our work can be consulted in the "Frameworks" section III-B.

For consistency and reproducibility purposes all experiments were made in a single machine whose specifications are shown in the table below.

Table I
Benchmark Machine specifications.

CPU	AMD Ryzen™ 7 1700 Processor 3.0GHz
RAM	16GB DDR4
GPU	Nvidia GeForce 1070

A. Dataset Preprocessing

In order to make the images from the [Tan+21b] dataset ready to be used the models developed, we needed to address the fact that all the images had different dimensions. This fact made it unfeasible to feed the images as is directly to a CNN and required us to reshape all the images in dataset. After a careful look at the dataset, we decided that we would work with images with dimensions 224x224 pixels and proceeded to proportionally re-scale all the images to said dimensions. In the case of the images which a re-scaling event would result in the image being shrunk down, the regions left empty were filled with a black strip, therefore minimizing any interference in the model's predictions.

B. Frameworks

In terms of frameworks, in this project besides all the standard python libraries, we used the following frameworks:

- **Pillow** [Cla15]: For loading the dataset images.
- **TensorFlow** [Aba+16]: For CNN model construction.
- **modAL** [DH]: For the AL cycle implementation.
- **openCV** [Bra00]: For image preprocessing (e.g re-scaling/padding).

IV. Experimental Setup

In this section we present the experimental setup for studying the effectiveness/performance of and active learning approach on pedestrian semaphore classification. In the first sub-section IV-A we discuss the implementation made in the scope of this project and in the second IV-B we move on to talk about the design of the experiments / experimental scenario.

A. *Implementation*

For the purposes of work and in order to implement the active learning cycle we needed well designed models that would allow us to observe the active learning contribution.

First of all, we started by implementing some state of the art CNN's since these are well known models and have proved to yield a good performance in image classification, therefore serving as the control group of our experiments. That being said, the state of the art CNN's we ended up choosing for implementation were LeNet [Lec+98] and the AlexNet [KSH12] whereas both are similar but architecturally one CNN has more depth than the other.

Afterwards, in order to later establish further comparison with the most recent work (to our knowledge) on personal companions for impaired pedestrians [Tan+21a], we decided to implement one of the CNN's used in said work. With this, our main goal was not only compare the performance with state of the art CNN's (control group), but also later check if the AL approach would better the performance obtained by this CNN that was custom built for this particular problem. The architecture of the CNN as previously discussed in the state of the art section (II), that at a very high level, is comprised of 5 convolutional layers followed by 3 fully connected layers. Throughout our work we will refer back to this CNN as *FlyNet*.

After the models were implemented it was time to introduce the active learning cycle into the models. For this work we decided to use uncertainty based sampling, implementing the multiple sampling strategies in order to assess which one was the best fit for our goals.

B. *Experimental Scenario*

Having the dataset loaded and all the necessary components of the AI system implemented we proceeded to testing it. The experiences conducted were designed to test the multiple models implemented as well as how they behaved when the active learning loop was introduced. Be that as it may, we ran each CNN model with 30 different seeds (from seed with number 0 to 29) with a training duration of 25 epochs, hence obtaining representative results of the performance of the CNN and avoiding a potential bias that running with the same seed might have introduced. The choice for 25 epochs training duration was due to the accuracy stabilization observed after 15 epochs, also in the [Tan+21a] paper they used the same parametrization which inclined towards us using this value.

Afterwards, we tested the active learning loop on each of the trained models using the multiple query strategies implemented (IV-A). As the parametrization of the the AL loop, we used an initial size of the training data of 600 known labels, a query size of 60 labels and, as the maximum number of queries made in the active learning loop, 11 queries, having concluded a total of 8 queries per learning strategy until the pool of unlabeled data was depleted. Furthermore, after each query, the model was retrained with 25 epochs in order to be able to learn the new examples provided. Last but not least, on the experiments made the considered metric was the accuracy obtained by the model using AL when compared with the model without

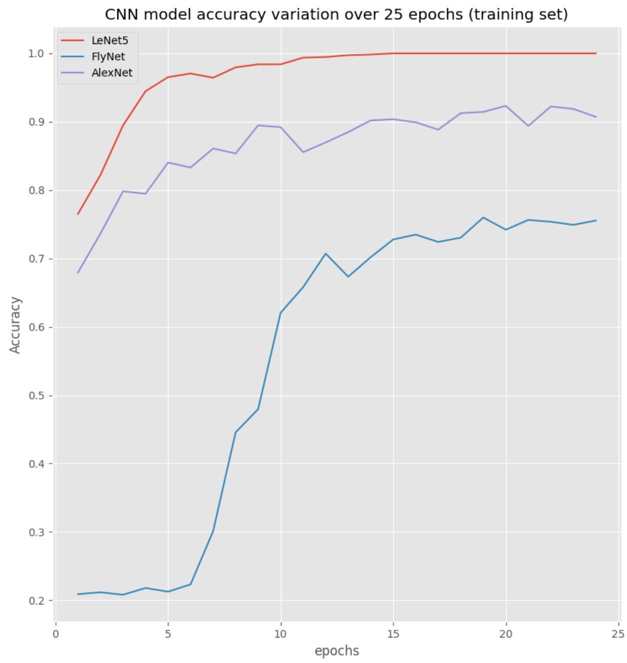


Figure 4. Accuracy evolution over 25 epochs for the multiple CNN's; training set results; seed 42.

it. Our main interest is to observe the impact in terms of performance (accuracy) and its evolution, e.g number of queries needed to obtain results the match the original model without the AL cycle.

V. RESULTS

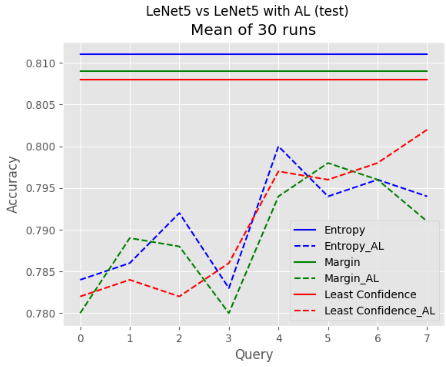
We started by looking at the performance of the models when considering the raw models without the active cycle implemented into it. The results obtained can be observed in the following table:

Table II
AVERAGE CNN'S ACCURACY OVER 30 RUNS (TEST SET)

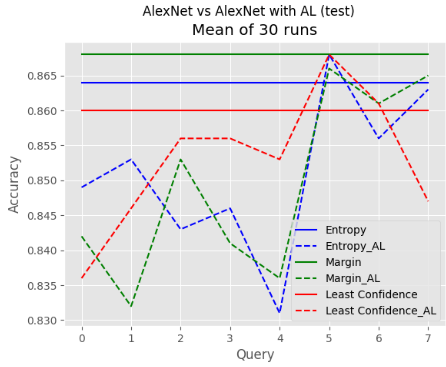
CNN	Accuracy
LeNet5	0.809
AlexNet	0.868
FlyNet	0.417

The results (table II) show that both LeNet5 and AlexNet show good performances on the testing set, unfortunately the FlyNet network implementation performs poorly, which is something unfortunate taking into account the accuracy reported by the authors of the CNN [Tan+21a] on their tests. These results on the FlyNet may be implementation dependent but the original authors do not provide any details about that so we can reproduce their work, leaving us to establish our analysis strictly with our implementation which yields this performance. Nonetheless, the results may still be used to assess the quality of the AL methodology we wish to test.

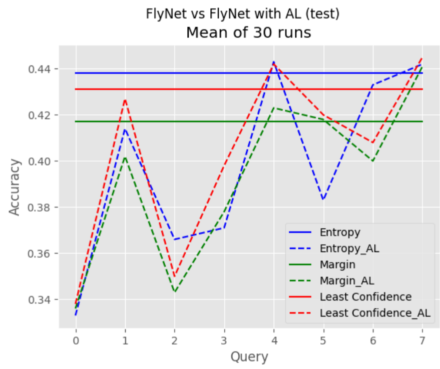
In the following graphics we show the comparison between the performances obtained when comparing the results obtained from running, on the test set, the raw CNN's against the results obtained when the AL cycle (with multiple query strategies) was introduced (figure 5). We also show the results obtained when comparing the performances obtained by the CNN's with AL, in the training set against the ones obtained in the testing set (figure 6).



(a) LeNet5

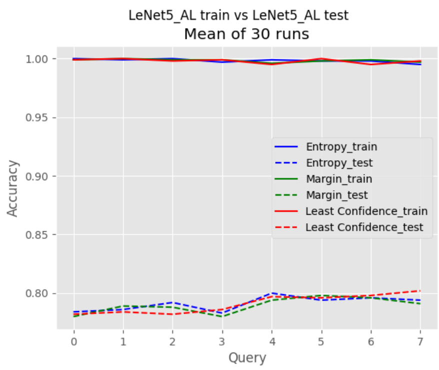


(b) AlexNet



(c) FlyNet

Figure 5. Performance evolution for the CNN's: LeNet5 (figure 5a), AlexNet (figure 5b), FlyNet (figure 5c), compared with CNN's + AL with different query strategies (entropy sampling (blue), margin sampling (green), least confidence (red) sampling).



(a) LeNet5

(b) AlexNet

(c) FlyNet

Figure 6. Performance evolution and differences between training and testing sets for the CNN's + AL: LeNet5 (figure 6a), AlexNet (figure 6b), FlyNet (figure 6c), with different query strategies (entropy sampling (blue), margin sampling (green), least confidence (red) sampling)

VI. Discussion and Conclusions

As we can observe from the results we obtained, in general the accuracies we got from using the CNN's with AL are very close to the ones obtained when using the raw CNN's with no AL, using a smaller amount of instances. This is visible on the multiple plots on figure 5.

When comparing the performance difference between the training set and the testing sets, figure 6, we can see that the results obtained are worse when we look at the test set in comparison to the training set. This is to be expected since the testing set contains data that the model has never seen, which leads it to have less accuracy. Nonetheless, the accuracy difference between training/test sets for the AlexNet and FlyNet CNN's is not blatant, showing that these models have great generalization capabilities, therefore being able to classify new samples accurately. In the case of the LeNet5, the results show that this CNN's is most certainly overfiting on the training data. This is evident since, with the training set the model is able to obtain accuracies that round 1.0, but when it is exposed to the test set it is only able to reach accuracies of about 0.80.

In conclusion, we are confident that the results obtained are representative of the improvements that an AL approach for training classifiers may bring, in the sense that we can train the models with less data. When comparing to state of the art methods, and the model in the [Tan+21a] paper, results also show that this methodology is really competitive.

VII. Future Work

From our research, we realized that there is still a lot to be done in this field. In fact, the development of assistive systems for impaired pedestrians is a fairly recent topic from the point of view of AI research, much so that with our work, we only touched a small topic which is the improvement of semaphore classification techniques for these systems by using active learning techniques. We feel that two main things must be done in the near future in order to enable more research in this area. First of all the creation of more datasets containing a wider variety of obstacles that a pedestrian may encounter is of paramount importance. The existence of more datasets will allow the same methodology discussed in this work to be used in the classification of more objects, therefore allowing for the development of a more capable and complete assistive systems. Last but not least, regarding our work in particular, the experimentation with other active learning query strategies is an important topic that we leave as future work, as other methodologies may be able to achieve even better results.

References

[Lec+98] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[Bra00] G. Bradski. *The OpenCV Library*. 2000. URL: https://pypi.org/project/opencv-python/.

[Set09] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin–Madison, 2009.

[Set11] Burr Settles. "From Theories to Queries: Active Learning in Practice". In: *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*. Ed. by Isabelle Guyon et al. Vol. 16. Proceedings of Machine Learning Research. Sardinia, Italy: PMLR, May 2011, pp. 1–18. URL: https://proceedings.mlr.press/v16/settles11a.html.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

[Cla15] Alex Clark. *Pillow (PIL Fork)*. 2015. URL: https://github.com/python-pillow/Pillow.

[Aba+16] Martín Abadi et al. *Tensorflow: A system for large-scale machine learning*. 2016. URL: https://www.tensorflow.org/lite/guide/python.

[Tan+21a] Haobin Tan et al. *Flying Guide Dog: Walkable Path Discovery for the Visually Impaired Utilizing Drones and Transformer-based Semantic Segmentation*. 2021. arXiv: 2108.07007 [cs.CV].

[Tan+21b] Haobin Tan et al. *Pedestrian Vehicles and Traffic Lights Dataset*. Sept. 2021. URL: https://drive.google.com/drive/folders/1UFcr-b4Ci5BsA72TZWJ77n-J3aneli6l.

[DH] Tivadar Danka and Peter Horvath. "modAL: A modular active learning framework for Python". In: (). available on arXiv at https://arxiv.org/abs/1805.00979. URL: https://github.com/modAL-python/modAL.