Faculty of Mathematics
and Computer Science

Aligned multi-task deep semi-supervised learning

Comănac Dragoș-Mihail

dragos.comanac@stud.ubbcluj.ro

Abstract

Computer vision has emerged as a key field of artificial intelligence because it aims to replicate functions of the human visual cortex. More specifically, computer vision aims to extract information from visual data such as images or videos. For example, object detection is one of the main computer vision problems in which instances of objects must be located and classified in visual data.

Deep convolutional neural networks became a standard that enabled reliable learning based data analysis, and as such, they are the key factors that turned computer vision into a very relevant field of research in our modern society.

The main purpose of this paper is to propose a research project in which a multi-task neural network is to be developed. The novel features that are to be research are task alignment and semi-supervised learning. The aim is to use this network on images in order to provide information in an assistive technology specially designed for visually impaired people. We also note that the possible insights learned during this project can be applied in many other scenarios or domains.

Keywords:
Multi-task; Deep learning; Semi-supervised learning; Task alignment

1. Introduction

Approximately 200 million people globally have moderate to severe distance vision impairment or are blind due to cataracts or uncorrected refractive errors, according to the World Health Organization [9]. This is out of a total of 2.2 billion people with vision problems. These individuals may struggle with daily tasks, but assistive technologies can help them maintain or improve their capabilities. Computer vision, which allows computers to substitute for visual functions, has the potential to be a main part of assistive technology for visually impaired persons (VIPs). It can help VIPs better understand their surroundings and perform various tasks.

Computer Vision, a branch of artificial intelligence and deep learning, enables computers to process and understand the content of digital images and videos. It can be seen as a way to mimic the functions of the eye and the human visual cortex, the part of the brain that processes visual information. While modern cameras have effectively replicated the functions of the eye, replicating the functions of the visual cortex has proven to be more challenging. However, current AI models are capable of performing well on specific tasks, and in some cases even surpassing human recognition abilities.

Computer vision has the potential to be a key component of assistive technology for VIPs by helping them to better understand and navigate their environment while performing various tasks. This technology can also help to prevent accidents. Additionally, mobile phone technology is cheap and easy to use, making it more accessible than specialized devices that may not be available anywhere on earth. Given the number of people with visual impairments and the ability of computers to substitute for visual functions, computer vision has the potential to significantly improve the lives of VIPs.

The ability to move freely is a fundamental human need, but VIPs often face difficulties and social exclusion when using transportation due to a lack of accessibility options [8]. This paper proposes a research project that aims to develop a multi-task neural network using deep learning that is suitable for systems with limited resources. The ultimate goal is to deploy this neural network on a mobile device for use by VIPs to improve their mobility and reduce their reliance on inadequate accessibility solutions.

2. Placement in the general field

Our research project is highly oriented towards computer vision, more specifically, the learning methods in this field

2.1. Tasks

Regression and classification are two of the most basic tasks, that are not specific to computer vision, but to artificial intelligence in general and are related more to the output of the algorithm. For instance, a regression algorithm outputs a number in a continuous space, and classification outputs a number in a discrete space. Essentially, computer vision problems are based on these two basic tasks.

One such problem is object detection, which consists of locating and classifying objects in an image by producing a set of bounding boxes which contain both information about the spatial coordinates of the object, and information about it's type or class.

Optical character recognition (OCR) is another relevant field for our problem because it is the task in which text is extracted from the image. This can provide valuable information to the VIP.

Another important task is segmentation that aims to classify every pixel in an image.

2.2. Multi-task learning

White the aforementioned tasks can help a lot in conveying information about the environment by themselves, there is a much bigger potential in combining them mainly due to computational reasons.

The naive way to solve a composed problem of several tasks is to use different algorithms for each task and run them separately. While this technically does the job, it is not sufficient for a real-time mobile application.

There are various methods that can be used in the field of computer vision, such as hand crafted features, but in this context of big data, the most relevant methods are related to machine learning, and more recently to deep learning which is better suited to visual data. Through supervised learning, deeper and deeper neural network models are now able to learn the complex patterns found in large amounts of data, thus they are a good fit for solving computer vision problems.

The type of neural network that we are interested in falls into the category of supervised learning, which is a subfield of the more general area of machine learning. They are called deep neural networks because they have multiple hidden layers.

It is important to note that deep neural networks have applications in various other domains such as segmentation and some of the knowledge and intuitions could be transferred between areas of interest.

The right way to combine several tasks is to use a single deep neural network that can learn from them. The idea is that most computer vision tasks share at a basic level a lot of features. For instance, each task could use information about the various shapes fond in an image. Therefore it makes sense to extract this kind of information using a shared backbone, after which each task can specialize separately.

This is also known as multi-task learning in which several tasks are combined, sharing input images. This also comes closer to how humans learn, compared to single task learning.

3. Objectives

In order to have a better idea of what we are trying to achieve, we list the main objectives behind this work that show how it can have a great impact in real world use cases and can be extended to solve other problems other than VIP mobility.

3.1. Low computational cost

Firstly, the main objective is to propose an algorithm that does not require large computational resources. That is because the algorithm should run in an environment with very little resources, and in real time. The final aim is to deploy the algorithm on an accessible and cheap mobile device such as a phone or other assistive device with an incorporated camera.

Since as many people as possible should benefit from this research, we need to perform a study of existing options in terms of mobile devices. The idea is that we first need to know what is the final destination for the artificial intelligence model in order to have a more guided research.

As such, we aim to find what is the preferred mobile device technology among the VIP and try to optimize for it. Although, we mention that this is most important in the later stages of the development. Already the real time constraint drives the research towards algorithms specifically designed for speed. This study about what the VIP use is more useful for fine-tuning the final details of the algorithm such that it runs on the specific device.

3.2. State of the art research

Another important aspect is the study of the state of the art. In order to combine different tasks, we should first know what are the best algorithms that solve each task separately. Therefore, our objective is to conduct a systematic literature review of the most recent methods for solving object detection, segmentation and OCR.

This should help in various design choices regarding network architecture, and also in choosing the proper datasets.

3.3. Cooperation between tasks

Another key objective that we propose for this research project is to see how one of the latest innovations in object detection described in [3] could be applied in the context of multi-task learning.

The classical approach in multi-task learning is to attach each task layers at various points in the backbone and train them somewhat separately at the head level. But more recently, the authors of [3] proposed a novel approach that actively aims to align two separate tasks. Initially, this was done to make classification and regression work together in the context of object detection but we believe that this is something that can be extended. As such, we aim to research how this can be done in the context of multi-task learning.

3.4. Semi-supervised

Traditionally, computer vision tasks belonged to the supervised learning category of algorithms, but they require a lot of labeled data which can get very expensive especially for segmentation. A more viable approach is a semi-supervised one, meaning that the multi-task neural network would learn from both labeled and unlabeled data. In this way more data can be used to improve the model with minimal efforts.

3.5. Multi-task framework

In order to actually be able to conduct our research activities, we need to develop a framework which will allow us to create multi-task models and train them. We will also have extensibility in mind, because ideally, more and more vision task should be added in order to improve the amount of knowledge extracted from the images.

4. Reseach methodology

4.1. Dataset

The first step is probably also the most important one. In order to train any machine learning algorithm, a dataset must be gathered. Its quality is crucial because it determines what the learning algorithm can actually learn and how well it can learn it.

As we have mentioned, we aim to develop a semi-supervised learning multi-task neural network. This means that the dataset must have a special structure, in order to satisfy the semi-supervised algorithms. Basically, such a dataset must contain pairs of inputs and outputs or just inputs alone.

It is a good idea to have the inputs to be samples from the same distribution in order to have some patterns. These inputs can be anything ranging from simple feature vectors to images or sequences of words and usually it is relatively easy to find lots of samples for input.

The tricky part is finding the correct output. Obviously, the outputs must be representative for the input and must describe what it should be learned. This part of gathering the outputs is especially hard because it means that each input must be processed and labeled. These labels can be quite costly, especially if expensive hardware or expert people are used in the labeling process. For instance, medical data could be such an example. Also, cheap labels can be quite costly in the end if there are a lot of inputs. And in this era of deep learning, a lot of data is required for an algorithm to learn, which is a big problem from the labeling point of view. This need for an extensive amount of good labels is one of the main pitfalls of supervised learning. This is the main reason behind the research into semi-supervised learning because it does not need labels for all inputs. In this way, the neural network can learn from a larger set of images, both labeled and unlabeled.

When it comes to object detection, the inputs are images, and the output represent a set of bounding boxes corresponding to the target objects in the image. They need to contain information about the class of the object and about the location of the object in the image. A common way to describe the location is to give the coordinates of the upper left and bottom right corners in the image scale.

If the bounding boxes for object detection are relatively cheap and easy to find, in the case of segmentation things get complicated. Each pixel in the image must be classified, which gets harder the larger the image gets. Usually this kind of label is very rare and that is why semi-supervised learning could play an important role here as it was shown in [1].

OCR is more similar to object detection, but still more complicated because each textual character in the image needs to be labeled.

In order to have a qualitative dataset, it is important to have labeling conventions. For instance, it is important to clearly describe what is considered to be an object of interest. Otherwise, the learning algorithm might be confused. Also, the input needs to be clear, at least for humans, because if a human can't tell what is there, it would be difficult for the learning algorithm.

Also, a common practice is to split the dataset into several parts. Firstly, a training dataset is necessary. Basically, on this, the weights of the algorithm will be updated, meaning that the patterns that describe the data are extracted from this information. Beside a training dataset, a validation dataset is necessary. No learning occurs on this data. As the name suggests, it is used to validate the quality of the algorithm. A common use is to save the models that perform good on this dataset.

This need of several datasets comes from the problem of overfitting. The idea is that if the model is too complex it will learn too well the data and it does not generalise well to unseen data. If we have a separation, we can spot overfitting by comparing the loss or another metric on the training and validation sets. If the model overfits, a continous improvement can be observed on the training set, while on validation the performance plateaus. As such, various regularization methods can be applied in order to alleviate this issue. This is a complex problem on its own.

Sometimes a training and validation datasets are not enough. It can happen that the by setting various hyperparameters of the learning algorithm, the validation set is also overfit. Therefore a new dataset, called test dataset is used. The idea is that the final performance measurements are done on it because it is the only data that is not seen during optimization, and such it would provide an idea about the performance of the algorithm in a real life scenario with new data.

Before deep learning, a common choice was to have a larger proportion of the data dedicated to validation or testing such as 20-30%. This was when the datasets were smaller. But nowadays, if the dataset is large enough, this percents can be lower. For example, if the dataset contains millions of images, it may be enough to have only a fer percents dedication for validation and testing each. The intuition is that it is more important to have data on which the model can actually learn. This works best if the dataset splits are diverse enough, such that the test set is representative.

When it comes to the actual choice for the dataset, there are a lot of options. Firstly, for each task, a dataset must be gathered. If the labels allow it, tasks may share datasets. For instance, the Microsoft Common Objects in Context (COCO) dataset [7] is the standard that object detection methodologies use in order to be relevant in the field. It also contains segmentation masks. Another good candidate for segmentation is Cityscapes [2]. As the name suggests, it depicts urban images which would be quite nice for the purpose of VIP mobility.

4.2. Neural network model

The next step is to select the learning algorithm. There are various options here, but we are going to focus on neural networks, because they have the most potential due to the fact that they can learn directly from the data various patterns. Given the multi-task context, the naive way to proceed would be to have a separate neural network that can execute each task independently. The most obvious flaw with this design is the excessive computational cost.

A better approach to multi-task learning is to have a single neural network architecture that can handle all required tasks. It is a common practice to split the neural network architecture in three parts, as we have depicted in Fig. 1. Each section of this architecture is important and deserves comprehensive studies on their own.

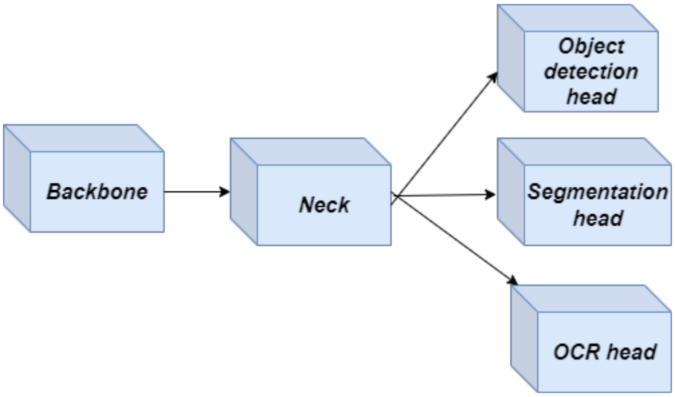

Fig. 1. Neural network architecture for multi-task overview

The backbone of a neural network is responsible for extracting feature maps from the input image that describe shapes and other features that can help identify the location and class of potential objects or other types of features. The specific design of the backbone can vary depending on the precision and speed requirements, with smaller networks being used for speed and larger networks providing better precision. Custom design is possible, but when there is not enough data to support the desired network size, transfer learning is often used.

Transfer learning involves using a pre-trained network and replacing the last layers with a neck and head architecture specific to the target problem. This allows the network to reuse knowledge learned from the larger source dataset. The number of layers cut from the image classifier neural network depends on the similarity between the target and source datasets. The network is first trained with the backbone frozen and then trained with the backbone unfrozen and a low learning rate to fit the target dataset.

For example a ResNet like architecture [4] can be used if high precision is needed, or a MobileNet like architecture [5, 10] is suitable if speed is of the essence.

The neck, which is an optional component, serves as an intermediary between the backbone and the head. Its main function is to further refine the features from the backbone to provide more information for the head. One example of a neck architecture is the Feature Pyramid Network [6], which utilizes multiple feature maps extracted from the backbone to allow the network to ”see” at different resolutions. The neck helps to provide the head with more detailed and accurate information for image analysis.

An important point of this research project is to study various ways to implement the three proposed heads in Fig. 1.

We want to focus on the issue of aligning tasks. Object detection involves two smaller tasks: classification and localization. One approach to solving these tasks is to use deep neural networks with two separate branches for classification and localization. However, this method has some drawbacks. One main disadvantage is that the classification and localization tasks do not work together effectively, leading to situations where bounding boxes with accurate localization may be eliminated during non-maximum suppression due to low scores from the classification branch.

Considering these arguments, Task-aligned One-stage Object Detection (TOOD) [3] proposes to actively align the two tasks. In this way, a if a predicted bounding box has a good confidence score, it should have a good localization also.

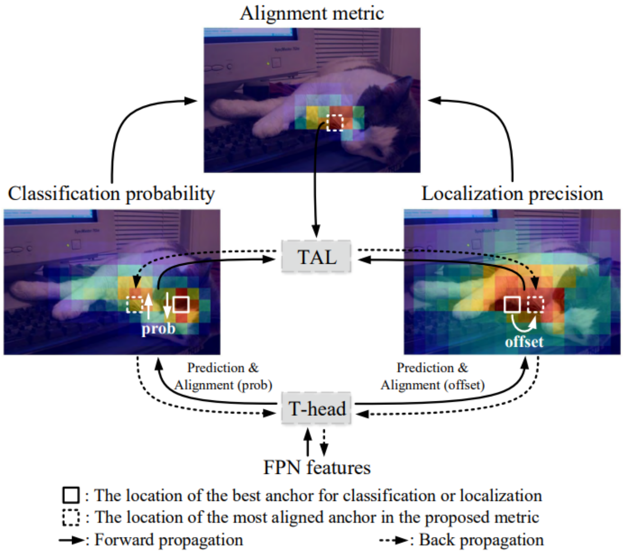
Alignment metric
Classification probability
Localization precision
TAL
prob
offset
Prediction & Alignment (prob)
Prediction & Alignment (offset)
T-head
FPN features
□ : The location of the best anchor for classification or localization
⬚ : The location of the most aligned anchor in the proposed metric
→: Forward propagation
⇢: Back propagation

Fig. 2. TOOD learning [3]

Fig. 2 illustrates the two components that help to achieve alignment of tasks. The first component is the task-aligned head (T-head), which can be integrated into any object detector using the backbone-neck-head architecture. This makes it a versatile solution. The other component is task alignment learning (TAL), which is not dependent on the T-head and can be applied to other object detectors, regardless of whether they use anchor-based or anchor-free approaches.

Having these arguments in mind, we aim to see how this aligment can be extended in the context of multi-task learning.

4.3. Training

Training a multi-task neural network might prove to be a bit trickier to train than a regular neural network. This is because, even though the various task might share some features, such as shapes, they still might conflict at some point.

There are various ways to train a multi-task neural network. One would be to take each head, and freeze the others, then train it with the backbone. But by doing this, the knowledge gained by one task might be broken by another.

The better way to do it is to train all heads at once, but a key point is to balance the loss that is propagated to the backbone. For example, some task might have a higher weight than others. An interesting approach here is to let the model learn those weights.

After this step, in order to better tune the heads to their specific tasks, each heads should be trained separately, but what is different from the first approach is that now the backbone is frozen. In this way, the tasks do not conflict with each other, and the knowledge in the backbone is not broken.

Another important aspect for training is data augmentation. This is a technique used to artificially increase the size of a dataset by creating modified versions of existing data. This is often done in machine learning to improve the performance of a model, especially when the original dataset is small or imbalanced.

There are many ways to perform data augmentation, including adding noise to the data, such as Gaussian noise or salt and pepper noise, rotating, scaling, or cropping images, flipping images horizontally or vertically, adjusting the brightness or contrast of images, or generating new data by combining multiple existing data points in different ways.

By creating these modified versions of the data, the model can learn to be more robust and generalize better to new data. However, it's important to be careful when using data augmentation, as it can also introduce bias or distort the original data distribution if not used properly, which would hinder the model capability to learn. The goal of data augmentation is also to keep the same meaning of the initial data.

4.4. Optimization

Another important step is optimization. In order to run a neural network on a mobile device with low resources, ideally the network should be optimised by using quantisation.

Neural network quantization is the process of reducing the precision of the weights and activations of a neural network model in order to reduce its memory footprint and improve its computational efficiency.

Various methods for quantizing a neural network have been introduced. Weight quantization involves reducing the number of bits used to represent the weights of a model. For example, instead of using 32-bit floating point values to represent the weights, we can use 8-bit integers. This can significantly reduce the model size and speed up computation, but it can also introduce some loss of accuracy.

Activation quantization reduces the precision of the activations (i.e., the output of each neuron) in the model. Activation quantization can be done in a similar way to weight quantization, by reducing the number of bits used to represent the activations.

Hybrid quantization combines weight quantization and activation quantization to further reduce the model size and improve efficiency.

As such, quantization is often used in production environments to deploy neural networks on devices with limited memory or scarce computational resources, such as mobile devices. It can also be used to speed up the training process by reducing the amount of data that needs to be processed. However, quantization can also introduce some loss of accuracy, so it's important to carefully evaluate the trade-off between accuracy and efficiency.

4.5. Evaluation

If the tasks were very interconnected during training, during the evaluation phase they are independent. Each task may be evaluated according to its specific metrics. The important thing is to use well known metrics that are used also by other researchers that worked on the same data. This would be useful in order to compare the results. Also, probably most importantly, the metrics should reflect the true performance and give valuable insights about the results. Ideally, flaws in the implementation can be detected by using good metrics.

5. Details about managing the project

5.1. Team structure

We expect that this project requires a specialist that should handle the data. Also, regarding the network architecture, someone that would handle the design of the backbone and neck and the overall multi-task architecture is needed. Then for each of the tasks, one researcher which would handle the specific task.

Therefore, overall the team would be composed of about five members. The team should be oriented towards a self organizing approach, but we believe that the researcher tasked with the multi-task architecture should have the most experience and be designated as a leader.

5.2. Dissemination

Dissemination of a research project refers to the process of sharing the results and findings of the project with others. This can be done through a variety of channels, including publishing research papers in academic journals, presenting findings at conferences or workshops, and sharing results on social media or through other online platforms. It is important to disseminate research results in order to share knowledge and contribute to the academic community, as well as to potentially impact policy and practice in the field.

Firstly, we need to identify the appropriate channels for dissemination. There are several options such as academic journals, conferences, workshops, and online platforms, but we think that the most suited for this domain are the conferences. The research results should also ideally be publicly available on some platform.

5.3. Budget

We expect that this project would last for around one year, with the possibility to extend it afterwards. In computing the budget we will only make predictions for one year.

First of all, a proper developing environment is essential. In general, neural networks are notorious for needing a lot of GPU power. One option would be to buy a lot of GPUs but this would be too expensive and they would be outdated in a few years. A better option is to use an already available option such as Google Colab. This is a service specially built around AI computing. It offers several tiers, ranging from free ones to very powerful ones. We opt for the Colab Pro+ which costs around 47 euros per month. This means that over a full year this would cost $564 = 47 \cdot 12$ euros.

For the team members we define two types of salaries. Because the team leader is required to have more expertise, he should have a net salary of 3000 euros per month, which means a 4719 base pay. We estime about 2000 euros for the other team members, which means a 3146 euros base pay. In total, over 12 months, this would add up to $94380 = (4719 + 3146) \cdot 12$ euros.

This kind of work can be done remotely, on any laptop (because the development is mainly on the cloud), so no costs regarding infrastructure. Also, there are quite a few good open source datasets, so no costs here either.

Ideally, a research paper would be resulted from this one year of research, which would be presented at a conference. Here, we estimate a buget of about 1000 euros for each team member, but this could vary. Thus $5000 = 5 \cdot 1000$ euros for dissemination.

In total, the research project would need a buget of $99944 = 564 + 94380 + 5000$ euros.

6. Conclusions

In conclusion, we have detailed most important parts of this research project, but the work is far from done. There are still many details that need to be straightened out that can't be planned this far in advance. Looking towards the future, we hope that this work could have a real world impact, with uses not only in assistive technologies for the VIP, but also in other use cases.

References

[1] Chen, X., Yuan, Y., Zeng, G., Wang, J., 2021. Semi-supervised semantic segmentation with cross pseudo supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2613–2622.

[2] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[3] Feng, C., Zhong, Y., Gao, Y., Scott, M.R., Huang, W., 2021. TOOD: Task-aligned One-stage Object Detection, in: IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society. pp. 3490–3499.

[4] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep Residual Learning for Image Recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[5] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CoRR abs/1704.04861.

[6] Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125.

[7] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft COCO: Common Objects in Context, in: European conference on computer vision, Springer. pp. 740–755.

[8] Low, W.Y., Cao, M., De Vos, J., Hickman, R., 2020. The journey experience of visually impaired people on public transport in london. Transport Policy 97, 137–148. URL: `https://www.sciencedirect.com/science/article/pii/S0967070X19308364`, doi:`https://doi.org/10.1016/j.tranpol.2020.07.018`.

[9] Organization, G.W.H., 2019. World report on vision. World Health Organization Publications , 77.

[10] Sandler, M., Howard, A.G., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. IEEE/CVF Conference on Computer Vision and Pattern Recognition , 4510–4520.