Turkish Journal of Science & Technology
18(2), 425-433, 2023

Research Paper
https://doi.org/10.55525/tjst.1266144

Computer Vision Based AutoML Platform

Burak ŞAHİN[1], Aytuğ BOYACI[2*]
[1] ATASAREN, Milli Savunma Üniversitesi, İstanbul, Türkiye
[2] Bilgisayar Mühendisliği Bölümü, Milli Savunma Üniversitesi, Hava Harp Okulu, İstanbul, Türkiye
[1] bsahin215@gmail.com, [*2] aboyaci@hho.msu.edu.tr

(Geliş/Received: 16/03/2023; Kabul/Accepted: 11/08/2023)

Abstract: The rapid increase in data production, thanks to technological developments and scientific research, leads to the development of Machine Learning (ML) and similar new data analysis tools. It was announced that Amazon Web Services (AWS), a cloud service provider, stored 500EB of data in 2021 [1]. ML is an alternative to traditional engineering methods and does not require field knowledge of the problem to obtain a solution. However, the implementation of ML Algorithms can be complex depending on the content of the data set, and expert knowledge is the most important factor to use these algorithms effectively. Various methods have been developed to find a solution to this problem. There are many different areas and problems that machine learning can be applied to. We have limited our research to problems that can be solved using computer vision and AutoML. We have used AutoML and computer vision-based solutions to solve object classification, detection and segmentation problems. Our goal is to develop a platform that will work without the intervention of any expert. Users can load their datasets, choose the method they want, and train their models according to the problem they choose without any other intervention. After the training process is over, they can use their models in real time by transferring them over the platform in real time with their own hardware.

Key words: AutoML, Computer Vision, Deep Learning, Machine Learning, Object Classification, Object Detection.

Bilgisayarlı Görü Tabanlı AutoML Platformu

Öz: Teknolojik gelişmeler ve bilimsel araştırmalarla sayesinde veri üretimindeki hızlı artış, Makine Öğrenimi (ML) vb. yeni veri analiz araçlarının geliştirilmesine neden olmaktadır. Bir bulut servis sağlayıcısı olan Amazon Web Hizmetleri'nin(AWS) sadece 2021 yılında 500EB'lik veri depolandığı açıklandı. ML, geleneksel mühendislik yöntemlerine bir alternatiftir ve çözüm elde etmek için sorunun saha bilgisini gerektirmez. Bununla birlikte, ML Algoritmaları uygulanması veri setinin içeriğine göre kompleks olabilmektedir ve bu algoritmaları etkin bir şekilde kullanmak için uzman bilgisi en önemli etkendir. Bu soruna çözüm bulmak için çeşitli yöntemler geliştirilmiştir. Makine öğreniminin uygulanabileceği birçok farklı alan ve sorun bulunmaktadır. Çalışmada bilgisayarlı görü ve AutoML kullanılarak çözüm elde edilebilemek hedeflenmiştir. Bu anlamda çalışmada obje sınıflandırma, tespit etme ve segmentasyon sorunlarını çözmek için AutoML ve bilgisayarlı görü tabanlı çözümler kullanılmıştır. Hedefimiz, herhangi bir uzmanın müdahelesi olmadan çalışacak bir platform geliştirmektir. Kullanıcılar verisetlerini yükleyip, istedikleri yöntemi seçip ve başka hiçbir müdahale de bulunmadan seçtikleri sorun özellinde modellerini eğitebilemektedirler. Eğitim süreci bittikten sonra, kendi donanımlarıyla gerçek zamanlı bir şekilde platform üzerinden aktarım yapıp modellerini gerçek zamanlı bir şekilde kullanabilmektedirler.

Anahtar kelimeler: AutoML, Bilgisayarlı Görü, Derin Öğrenme, Makine Öğrenmesi, Nesne Sınıflandırma, Nesne Tespit Etme

1. Introduction

Today we are observing huge increase in the data generation with the developments of new technologies and the new scientific research [1]. The acceleration that observed on the data generation, leads to new advancement to new data analysis tools, which aim to examine the data and obtain meaningful results form the data. Machine Learning is the one of those tools [2]. Machine Learning is developed as an alternative for conventional engineering methods [3]. The main difference between machine learning and conventional engineering approaches are to obtain solutions to the problem, it doesn't require to field knowledge of the problem. The solution can be reached by using the data set obtained directly from the problem. Although this method more succesful than hand crafted methods, it is not perfect. In Prediction Based Modelling Methods, there is not a direct approach for usage of machine learning algorithms. This situation increases the dependencies on the expert's knowledge about these algorithms. Because there is a wide variety of algorithms and problem types. Development of the machine learning algorithms and applicability to many different fields at the same time brings the need of increasing numbers of experts about these algorithms. The knowledge of experts in these algorithms, significantly affects the success of algortihm. Since experts are not always sufficiently and equally competent, various methods have been developed to automate these algorithms in a way that requires minimum expert intervention.

Machine learning pipeline has consist of six different stages. This phases in respectively; data gathering, data cleaning and feature engineering, choosing a model that suitable with dataset, hyperparameter configuration,

training and evaluation. At each of these stages, there are required various adjustments processes by the expert and because of this adjustments success of the model directly affecting by the knowledge of the expert. Maybe the most important of this stages, hyperparameters adjustments and developing a suitable model for the dataset.

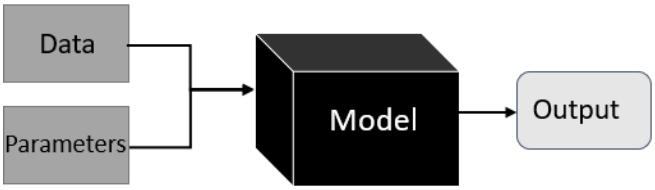

Figure 1. Black Box Model [4].

As seen in **Figure 1**, machine learning models also can be named as Black Box Models. The reason for this "that is as systems that hide their internal logic to the users" [4]. If we need to pharese in different way "Furthermore, the models learn from artificial datasets, often with bias or contaminated discriminating content" [5]. These models are given the dataset for training and the hyper-parameters that will be used during the training of the model for the developed model and cannot be changed by the model during the training. Since each of these stages is a separate subject of expertise, poorly trained models may lead to incorrect results due to the wrong model and hyperparameter selection. It is not always possible to find experts to develop these models and adjust the hyperparameters according to the distribution, quantity, and diversity of the available dataset. This situation leads to birth of a new approaches like Combined Algoritm and Hyper-parameter Selection(CASH). As given in the Equation 1, CASH Approach is aiming to get best model and the hyperparameters for the model with testing different algortihms($\mathbf{A} = \{A^1, A^2, A^3\}$) with different hyperparameters($\boldsymbol{\theta} = \{\theta^1, \theta^2, \theta^3\}$) on the same problem.

$$A^*, \theta^* = \text{argmin}_{A \in A, \theta \in \theta} E\left[L_\tau\left(A_\theta(D_{\text{Train}})\right)\right] \quad (1)$$

Deep Learning is a subfield of Machine Learning. The major difference between them, in The Deep Learning models build as a iteratively stacked layers of architecture [6]. Before The Deep Learning has proven itself, SIFT was the state of art method used various application such as image matching and object detecting in the computer vision [7]. The work of Alex et al. maybe one of the biggest game changers in the Computer Vision [8]. After this work, researchers have begun to use Deep Learning more frequently in their research. As we mentioned before there are lots of different problems and algortihm variations for this reason in our work we only focused Computer Vision and in this field we choose the focusing object detection, classification and segmentation problems.

2. Related Work

When the AutoML field is examined, there are three generally accepted basic titles. These are Meta Learning, Neural Architecture Search and Hyperparameters Optimizations. Meta-learning is a technique where a machine learning algorithm learns from other algorithms that have been successfully trained and tested and working on the same or similar problem. Meta-learning was first introduced by J.Schmidhuber in 1987 under the title of "method that learns by referencing a group of learning" [9]. Since this date, the increasing amount of data and the increasing training time as a result of growing Deep Learning Algorithm Architectures have had a significant impact on the prominence of meta learning.

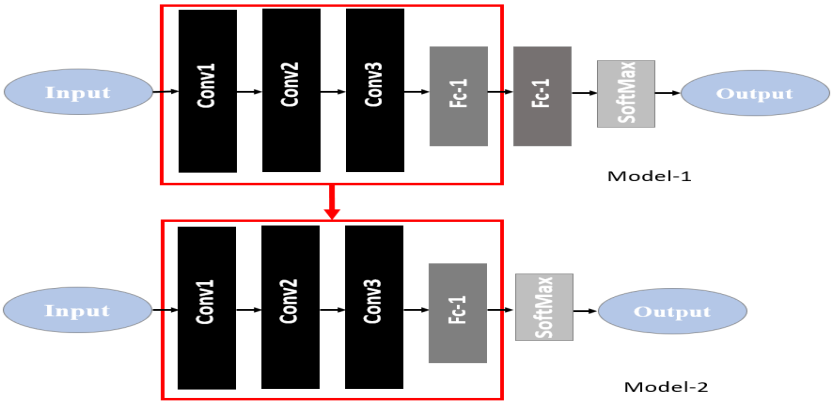

Figure 2. Transfer Learning.

To overcome these and similar problems, a number of new methods such as transfer learning and few-shot learning have been developed. Transfer learning. As can be seen in **Figure 2**, it provides the transfer of information between fields [10]. Instead of training a model from scratch to solve the current problem, we take the weights of another model that was previously trained to solve a similar problem, transfer it to model we want to use and customize it for our own needs. This approach not only shortens the model training time, but also allows us to obtain good results by fine-tuning in cases where the amount of data is low.

In a study in terms of AutoML and computer vision, it was done by Zeng Y. et al. [11]. In this study, thanks to the Google Cloud AutoML platform offered by Google as a service, they were able to train a model to detect breast cancer, which is mostly seen in women, by simply uploading the dataset.

In a study by Daniela M. et al., two separate models, SAI-G and SAI-C, using Google Cloud AutoML and Clarifai platforms; were trained to detect positive and negative emotions, and performance results were compared [12].

The MNIST Database was created by Yan L. et al. to set a standard for comparing the performance of Machine Learning Algorithms [13].

Jiancheng Y. et al. collected a similar dataset for 10 different cancer types and tested various AutoML approaches using AutoKeras, auto-sklearn tools and Google Cloud Vision Platform [14].

Intrusion is an important problem especially for countries with large borders and neighboring many countries. Countries with such geographical disadvantages set up walls and surveillance systems along their borders in order to ensure their border security. Because these systems are connected to a network, they suffer from a number of security vulnerabilities that every networked device suffers from. Intrusion Detection Systems have been developed to overcome these and similar problems. As in many fields, there are various Machine Learning applications in this field. In their study, Abhilash S. et al. developed an AutoML approach to select and train the most suitable machine learning algorithm for the Wireless Sensor Network's features [15]. Four features of the network were selected (the area covered by the network, the sensing distance, the transmission distance and the number of sensors), and a sensors data set was obtained by randomly generating these features within a certain range that can be found in real life. Then, they trained this dataset with the AutoML model they developed and determined the most suitable machine learning algorithm for the sensor at hand. Due to the complex data model space in AutoML techniques, researchers often limit their research to small models and data.

U-NET is a CNN-based image segmentation algorithm developed by Olaf R. et al. [16]. In the method section, the operation of this algorithm will be mentioned. On the other hand, Tonmoy S. et al. have tried to AutoML approach for a larger architecture such as the encoder-decoder architecture U-NET [17].

Up to this point, we have talked about a lot of AutoML studies. When it comes to machine learning, there is no universal AutoML System suitable for every problem that will meet everyone's needs. We have some methods available with a variety of tested successful results. However, even in this case, we need to be able to compare these systems in order to find out which system is the most suitable for the problem at hand. The study by Gijbers et al. provides us with a framework to compare AutoML systems [18]. This tool they developed and has been tested on five different AutoML algorithms and thirty nine diffe-rent datasets.

Erin et al. developed a distributed architecture platform called H2o [19]. The platform does many steps itself, including data preprocessing. However, H2o is limited to supervised learning only.

Evolutionary algorithms can be defined as “EA aims to find solutions to complex real-world problems using the simulated evolution method.” [20]. These algorithms improve the available solution population by mutating the individuals proposed for the solution of the problem and transferring information among themselves during a

certain iteration period. Using Evolutionary Algorithms, Liang et al. have developed an AutoML Framework that allows both to reduce the size of deep neural network architectures and to optimize hyperparameters [21].

Reinforcement learning(RL) "Helps agents communicate efficiently with their environment, enabling sequential decisions to be made" [22]. Agents are trained with the rewards and punishments they receive as feedback from their environment as a result of each decision they make. Mobile devices are widely used today. With the spread of these devices, they have started to apply various machine learning methods in the applications used in these devices. However, these devices, unlike desktop computers, do not have a stable energy source, and they also use smaller and less capable hardware. For this reason, especially deep artificial neural network etc. architectures need to be as compact as possible. Yihui et al., using the rl method in their study; They offer an AutoML Framework that will reduce the size of deep meural metworks for mobile devices [23].

Apostolos et al. developed an edge computing-based method for video analysis [24]. The AutoML method they developed adjusts the parameters of both the wireless sensor network and the Artificial Neural Network.

Artificial neural networks are added layer by layer on top of each other, and while the size of the architectures continues to increase, more successful but computationally costly models emerge. This leads to the emergence of various, specialized hardware such as deep learning accelerators that aim to train artificial neural networks faster. A deep learning accelerator can be defined as "Deep learning accelerators are considered as hardware architecture, which are designed and optimized for increasing speed, efciency and accuracy of computers that are running deep learning algorithms" [25]. In their study, Suyog et al. developed an AutoML approach that adjusts the parameters of the model by returning the loss value, taking into account the computational power of the accelerator [26].

3. Method

Our aim in the study is to enable users to make predictions with their models using their own cameras on the same platform in real time after they have uploaded the data to the platform in a pre-prepared format, selected the model suitable for the purpose they want, and trained their models, without having deep learning knowledge. The aforementioned structure is summarized in **Figure 3**. Before starting to explain our work we should set the limits of current status of our experimental platform design. Designing platform is a complex task. Our platform is limited to working with only image datasets which is structurally stored. In the computer vision there are lots of problem that to waiting for solution or optimization and improvement. As we try to build a complete platform and in order to demonstrate our platform strong sides, we have choosen from some proven solutions such as object classification, detection and segmentation.

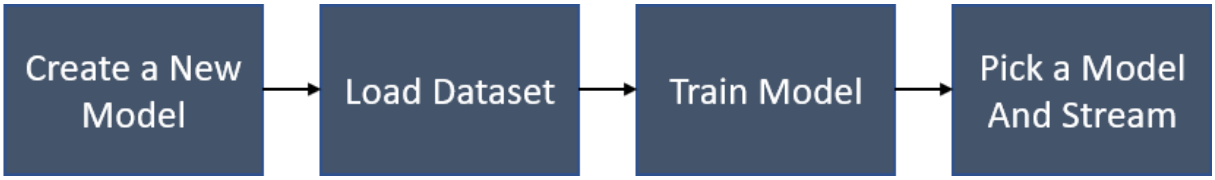

Figure 3. Main Event Loop

Convolutional Neural Networks(CNN) were first introduced by Yann LeCun et al. and it was used to detect single-digit numbers written in handwritten format [27]. Compared to classical Machine Learning methods, we need a large amount of data to train CNN. As the amount of data needed increases, the training time increases. These and similar problems have caused delays in the spread of CNN. A simple CNN Architecture is given in **Figure 4**. We use the same architecture in our own work while classifying objects. There is no feature that makes this architecture special. We chose this architecture because we wanted an architecture that would work fast and suitable for the generalized approach. We used the CIFAR-10 dataset to test our classification model [28]. The CIFAR-10 dataset consists of a total of 60,000 images of 10 different classes.

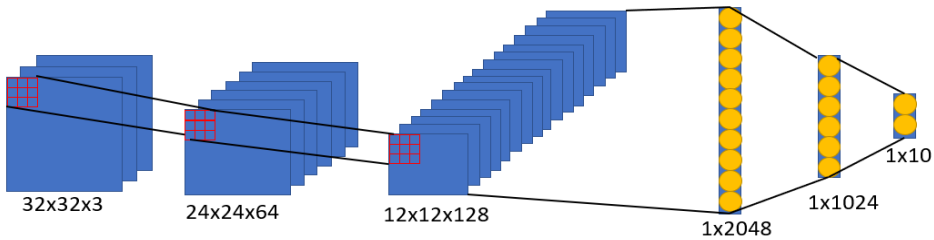

Figure 4. Basic Net

We used the Rectified Linear Unit(ReLU) activation function between the layers in the network we used for classification. ReLU entered the literature for the first time with a study by Fukushima K. [29]. The function is given in Equation 2. The feature that makes this function more successful than other activation functions is that it prevents the Disappearing Gradient Problem. "Disappearing Gradient Problem in back propagation stage, Convolutional Neural Networks calculate gradient using chain rule method. Multiplication of small numbers causes exponential reductions in gradients." [30].

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x > 0 \end{cases} \tag{2}$$

Our model consists of 3 convolutions followed by MaxPooling Layers. Convolution process; as seen in the red kernel in **Figure 3**, it is the process of moving a 3x3 kernel over the image and obtaining various feature maps from the image. The MaxPooling method is a process that we apply to reduce the dimensions of the obtained feature maps. The MaxPooling method selects the maximum value in a kernel and you get another feature map with a smaller size than the feature map to which it is applied [31]. Of all the pooling methods, we specifically choose this method because it focuses on the most important points in the feature map. The equation of the MaxPooling method is given in Equation 3.

$$\sum_{k=1}^{d} \max\{x_k, \dots, x_{k+4}\} \tag{3}$$

We preferred the Yolov3 algorithm for object detection [32]. The Yolo Algorithm was developed by Josep R. et al. [33]. Since Object detection algorithms such as Region Based Convolutional Neural Networks(R-CNN) do not create an artificial Neural Network structure from start to finish, the feature extractor layer in the first layer of the algorithm cannot be trained and improved with more data. Yolo algorithms, on the other hand, give better results as they are trained, as they create an entire Neural Network structure. There are newer versions of the Yolo algorithm that produce better results. However, we chose to use Yolov3 because problems such as hardware and server created a bottleneck for us. Yolo algorithms divide an image into bounding boxes(bboxes) of equal size. Each box is responsible for detecting the object falling into itself in the image. Yolo has a single detection layer of 13x13 in total, while Yolov3 as given in **Figure 5**, has three different sized detection layers, 13x13, 26x26 and 52x52. The reason why this structure is preferred is to facilitate the detection of small-sized objects. We used the COCO dataset to test our model [34]. The COCO dataset consists of 80 different objects and 330 thousand images in total.

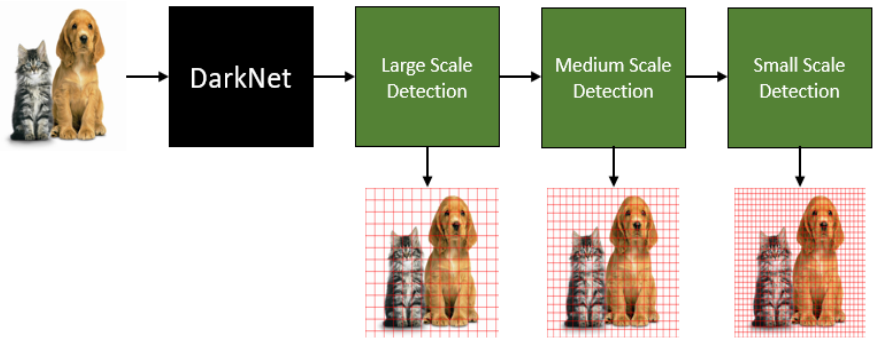

Figure 5. Top Level Yolov3 Algortihm

We need to talk a little bit about the Architecture of the Yolov3 Algorithm and how it works. Yolo algorithms basically consist of object detection layers added to the output of a pre-trained object classification algorithm. In the structure we use, DarkNet-53 is used as an object classifier. This architecture is a powerful architecture consisting of several residual net layers. The biggest innovation in Yolov3 is the addition of multiple object detection layers to the output of the object classifier. In this building, a method similar to the pyramid-shaped architecture made by Lin T. et al. was used [35]. It is the detection of small-scale objects by dividing an image into identical boxes of different sizes at different scales. Three anchor boxes with different geometric shapes are

assigned to each of these boxes. Anchor Boxes are "used as the first estimates of bounding boxes of many modern Deep Learning based object detectors" [36]. Anchor Boxes are predefined and loaded into the model. As a result, the model returns the coordinates, class and confidence score of the detected object, obtained from the boxes in 3 different detection layers. While testing the model we built, we trained 5 epochs and made sure it got the results we wanted. In order to speed up the testing process, we used the weights of the previously trained model by Joseph R. [37]. We mentioned that Yolo Algorithms get better as they are trained. When we say better, we are talking about both the accuracy and working speed of the algorithms. At the end of each iteration, it produces the result given in Equation 4 for three different scales. As given in Equation 4, b_x and b_y are the center of the bounding box, b_w and b_h are the width and length; p_c and c represent the probability of confidence score and the total number of classes.

$$y = (b_x, b_y, b_w, b_h, p_c, c) \tag{4}$$

Based on this result alone, we can see how costly an algorithm Yolo is. While training the Yolo algorithm, the p_c value in the bboxes it estimated over time gives more accurate results. Since the algorithms metric is p_c, which basically allows us to decide whether there is an object in the predicted bbox, the higher this value, the faster our process of evaluating the remaining bboxes is eliminated. **Figure 6** shows the running state of the Yolov3 model. In order to demonstrate working state of real time detection, we have decided to took totally 10 seconds of a time interval. We have divide this time interval equal three part with five seconds time lapse. In frame t = 0.0's our platform makes only one prediction with using trained Yolov3 model for person(green prediction box) on the front of the camera. In frame t = 5.0's our model makes three prediction for person, chair and window in orderly green, purple and pink prediction boxes. In t = 10.0's our model makes only one prediction for chair in the front of the camera. As we can see from last frame our model is not perfect. Frame t = 10.0's and t = 5.0's nearly identical but it's not able detect window as it did in frame t=5.0's. There are lots of improvement can be made for this situation for example; using more advanced yolo versions, some data augmentation techniques, more data or spending more time on training phase.

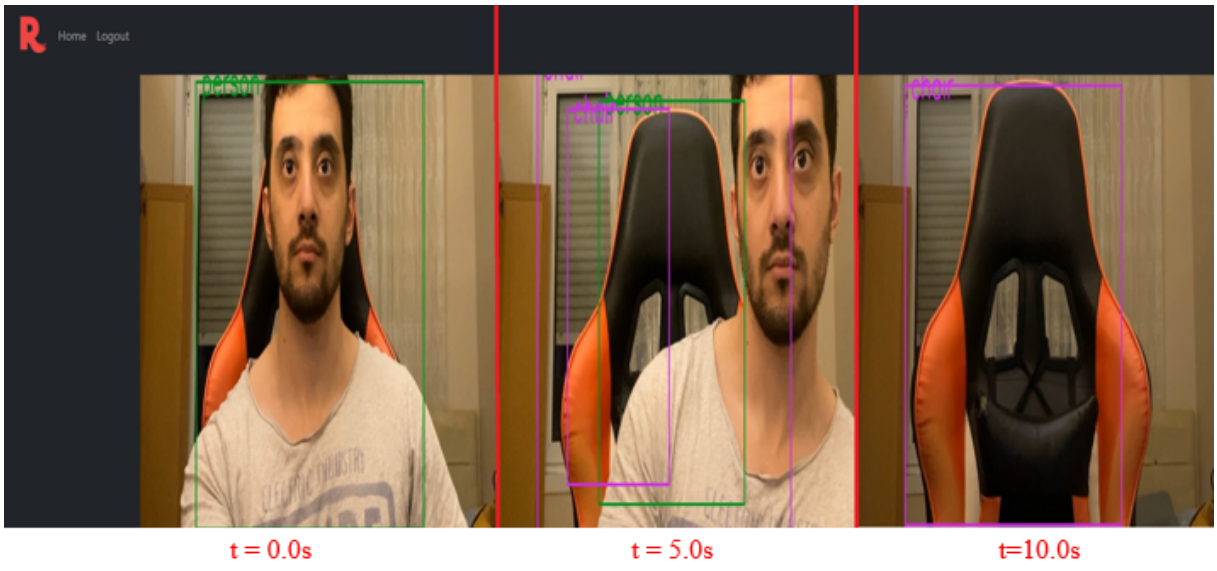

Figure 6. Object Detection On Working

Object Segmentation is mostly used in autonomous vehicles [38,39]. In its simplest form, segmentation is the process of assigning all pixels in an image to a class. What makes segmentation different from detection; While detecting only the relevant objects in the detection process, the segmentation process gives us more information about the structural features of the scene. Depending on the size and variance of the image at hand, the running speed of your algorithm is also affected.

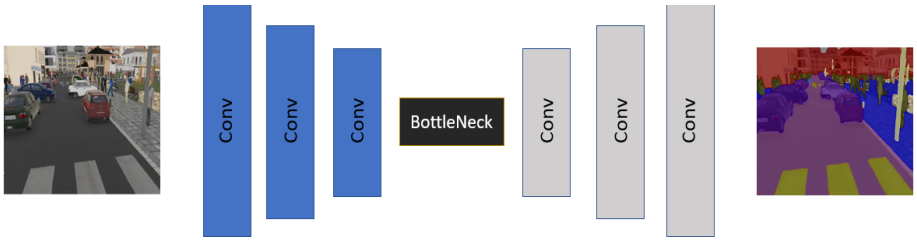

Figure 7. Encoder-Decoder Architecture Top View

There are many different types of architecture and approaches in the literature for Object Segmentation. Yuan et al. also tried a model using Transformers, which gave good results on the cityscapes dataset [40,41]. Yan et al. developed a model that examines the image at different scales, similar to Yolov3 [42]. Another Transformers-based study was done by Choi et al.[43]. Unlike the aforementioned studies, we preferred the previously mentioned U-Net model. These models we mentioned work more regionally. Our purpose in adding Object Segmentation to our platform is to gain more general information about what the scene is about. Therefore, as shown in **Figure 7**, U-Net, an Encoder-Decoder based architecture, was preferred. Encoders are used to compress information. They transfer the information they have compressed to the next layer. As information gets compressed, the size of the available data is getting smaller and smaller. Finally, the layer called BottleNeck is reached. The function of this layer is to keep the most important structural features necessary for us to reconstruct the data. The decoder, on the other hand, allows to regenerate the data based on the information in the BottleNeck layer. While Convolution Layers are used in Encoder Architecture, Convolution Transpose layers are used in Decoder. To define the work done by the Convolution Transpose layer, it "makes a transformation going in the opposite direction of a normal convolution" [44]. These two layers do the opposite of what each other does. While the layers learn the attributes of the data (color, brightness, etc.), the BotteleNeck layer learns the skeleton of the data. We used the Cityscapes dataset to test our segmentation model. This dataset consists of 20,000 images obtained from 50 different cities, with a total of 30 classrooms. We only used 19 classes in our study because we thought some classes were unnecessary. While setting up the models, we made use of another pre-installed ready models [45,46]. We had to make some performance improvements to make our model run faster when running in real time. One of these methods is to make an estimation by converting the image to a smaller size than its original size, and then converting the obtained segmentation map back to its original dimensions. Another method we apply is to convert the image to black and white. All these measures we have taken may not be a problem with better equipment. But we need to take these precautions for our current hardware.

4. Conclusion

We observed that the number of studies combining the field of computer vision with AutoML is very few in the literature. In order the demonstrate, these two fields can merge in a single product for solving object classification, detection and segmentation tasks, in our work we have try to design a platform that can train a ML model and stream the outputs of the model to the user without any expert intervention at all. We must warn the readers in this part, our platform is not production ready and unfortunately, we can't not host it at this stage. This work is an attempt to explore what can be done experimentally rather than a finished product. While we successfully handled the object classification and detection tasks, we had some difficulties with the segmentation task. The reason for this is that we prefer a computationally heavy model and method in order to obtain a generalizable model for segmentation and to obtain general information about the stage without disturbing the integrity of the stage. For object classification task, we have used BasicNet and achieved **%77** classification accuracy on Cifar-10 datasets. As we explained in method section, we had to use trained weights sets whis has **%55.3** mAP and **35** fps [37]. We did bulk of the computing in server side. This is not useful in delicate usage of real world environment (Traffic, Railway crossing). For the future work several improvements can be made for segmentation task and Edge Computing can be added, as Apostolos et al did.

Acknowledgment

This study was carried out within the scope of the thesis of the National Defence University, Atatürk Institute of Strategic Studies and Graduate Education. The idea stage of this study was carried out by B.Ş. and A.B., implementation phase carried out by B.Ş.

References

[1] Adadi A. A survey on data-efficient algorithms in big data era. *Journal of Big Data* 2021; 24: 8(1).

[2] Borgi T, Zoghlami N, Abed M, Naceur, MS. Big data for operational efficiency of transport and logistics: a review. In 2017 6th IEEE International conference on Advanced Logistics and Transport (ICALT) 2017; pp. 113-120.

[3] Simeone O. A very brief introduction to machine learning with applications to communication systems. IEEE Transactions on Cognitive Communications and Networking 2018; 4(4): 648-664.

[4] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM computing surveys (CSUR) 2018; 51(5): 1-42.

[5] Buhrmester V, Münch D, Arens, M. Analysis of explainers of black box deep neural networks for computer vision: A survey. Machine Learning and Knowledge Extraction 2021; 3(4): 966-989.

[6] Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, ... & Farhan L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of big Data 2021; 8: 1-74..

[7] Lowe DG. Object recognition from local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision 1999; 2: 1150-1157.

[8] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Communications of the ACM 2017İ 60(6): 84-90.

[9] Hospedales T, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence 2021; 44(9): 5149-5169.

[10] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, ... & He Q. A comprehensive survey on transfer learning. Proceedings of the IEEE 2020; 109(1): 43-76.

[11] Zeng Y, Zhang J. A machine learning model for detecting invasive ductal carcinoma with Google Cloud AutoML Vision. Computers in biology and medicine 2020; 122: 103861.

[12] Marcu D, Mirela D. Sentiment Analysis From Images-Comparative Study of SAI-G and SAI-C Models' Performances Using AutoML Vision Service from Google Cloud and Clarifai Platform. International Journal of Computer Science & Network Security 2021; 21(9): 179-184.

[13] Bottou L, Cortes C, Denker JS, Drucker H, Guyon I, Jackel LD, ... & Vapnik V. Comparison of classifier methods: a case study in handwritten digit recognition. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing 1994; 2: 77-82.

[14] Yang J, Shi R, Ni B. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI) 2021: pp. 191-195.

[15] Singh A, Amutha J, Nagar J, Sharma S, Lee CC. AutoML-ID: Automated machine learning model for intrusion detection using wireless sensor network. Scientific Reports 2022; 12(1): 9074.

[16] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference 2015; Part III 18: pp. 234-241.

[17] Saikia T, Marrakchi Y, Zela A, Hutter F, Brox T. Autodispnet: Improving disparity estimation with automl. In Proceedings of the ieee/cvf international conference on computer vision 2019: pp. 1812-1823.

[18] Gijsbers P, Bueno ML, Coors S, LeDell E, Poirier S, Thomas J, ... & Vanschoren J. Amlb: an automl benchmark. arXiv preprint 2022; arXiv:2207.12560.

[19] LeDell E, Poirier S. H2o automl: Scalable automatic machine learning. In Proceedings of the AutoML Workshop at ICML 2020.

[20] Vikhar PA. Evolutionary algorithms: A critical review and its future prospects. In 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC) 2016: pp. 261-265.

[21] Liang J, Meyerson E, Hodjat B, Fink D, Mutch K, Miikkulainen R. Evolutionary neural automl for deep learning. In Proceedings of the Genetic and Evolutionary Computation Conference 2019: pp. 401-409.

[22] Dridi S. Reinforcement Learning-A Systematic Literature Review 2022.

[23] He Y, Lin J, Liu Z, Wang H, Li LJ, Han S. Amc: Automl for model compression and acceleration on mobile devices. In Proceedings of the European conference on computer vision (ECCV) 2018: pp. 784-800.

[24] Galanopoulos A, Ayala-Romero JA, Leith DJ, Iosifidis G. AutoML for video analytics with edge computing. In IEEE INFOCOM 2021-IEEE Conference on Computer Communications 2021: pp. 1-10.

[25] Bolhasani H, Jassbi SJ. Deep learning accelerators: a case study with MAESTRO. Journal of Big Data 2020; 7, 1-11.

[26] Gupta S, Akin B. Accelerator-aware neural network design using automl. arXiv preprint 2020: arXiv:2003.02838.

[27] Forsyth DA, Mundy JL, di Gesú V, Cipolla R, LeCun Y, Haffner P, ... & Bengio Y. Object recognition with gradient-based learning. Shape, contour and grouping in computer vision 1999: 319-345.

[28] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images 2009.

[29] Fukushima K. Cognitron: A self-organizing multilayered neural network. Biological cybernetics, 1975; 20(3-4): 121-136.

[30] Shah A, Kadam E, Shah H, Shinde S, Shingade S. Deep residual networks with exponential linear unit. In Proceedings of the third international symposium on computer vision and the internet 2016: pp. 59-65.

[31] Gholamalinezhad H, Khosravi H. Pooling methods in deep neural networks, a review. arXiv preprint 2022; arXiv:2009.07485.

[32] Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint 2018; arXiv:1804.02767.

[33] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition 2016: pp. 779-788.

[34] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, ... & Zitnick CL. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference 2014; 6(12), pp. 740-755.

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

[36] Zhong Y, Wang J, Peng J, Zhang L. Anchor box optimization for object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2020: pp. 1286-1294.

[37] Yolov3 Weights Retrieved January 2, 2023 from https://pjreddie.com/darknet/yolo/

[38] Ma Y, Mosskull A, Xiang A. 3D Semantic Segmentation for Autonomous Cars.

[39] Zhang Z, Fidler S, Urtasun R. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016: pp. 669-677.

[40] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, ... & Schiele B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition 2016: pp. 3213-3223.

[41] Yuan Y, Chen X, Wang J. Object-contextual representations for semantic segmentation. In Computer Vision–ECCV 2020: 16th European Conference, 2020; 6(16): pp. 173-190.

[42] Yan H, Zhang C, Wu M. Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention 2022; arXiv:2201.01615.

[43] Choi S, Kim JT, Choo J. Cars can't fly up in the sky: Improving urban-scene segmentation via height-driven attention networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2020: pp. 9373-9383.

[44] Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning 2016; arXiv:1603.07285.

[45] UNet Model Retrieved February 10, 2023 from: https://github.com/hamdaan19/UNet-Multiclass

[46] Yolov3 Pytorch İmplementation Retrieved January 2, 2023: github. GitHub Retrieved from: https://github.com/eriklindernoren /PyTorch-YOLOv3