Detecting anomalous road traffic conditions using VGG19 CNN Model

M. Rajeshwari, and CH. MallikarjunaRao

¹MTech Student, Computer Science and Engineering, GRIET, Hyderabad, Telangana, India.

²Professor, Computer Science and Engineering, GRIET, Hyderabad, Telangana, India.

Abstract. Anomaly Detection on the real time road traffic has tremendous application possibilities in metropolitan road safety and traffic management. Due to the effect of numerous factors, for example: climate, viewpoints and road conditions in real-time traffic scene, Anomaly detection actually faces many difficulties. There are many reasons for vehicle accidents, for example: crashes, vehicle on flames and vehicle breakdowns, which exhibits distinctive and obscure behaviours. In this paper, we approached with a model to identify oddity in street traffic by monitoring the vehicle movement designs in two unmistakable yet associated modes which is 1. The vehicle's dynamic mode and 2. The vehicle's Static mode. The vehicle's static mode investigation is gained using the background modelling after the detection of a vehicle, this strategy is useful to locate the unusual vehicle movement which keep still out and about. The dynamic mode vehicle examination is gained from identified and followed vehicle directions to locate the strange direction which is distorted from the predominant movement designs. The outcomes from the double mode investigations are at long last fused together by driven a distinguishing proof model to get the last peculiarity. For this research we are using traffic-net Dataset, VGG19 CNN model along with ImageNet weights and OpenCV.

1 Introduction

An ever-increasing number of families presently have their own vehicles and going via vehicle has gotten an exceptionally normal and advantageous path in day-by-day metropolitan life. The street condition consequently gets extraordinary consideration from the general population. Terrible street conditions can make monstrous misfortune to the social economy, and compromise the individual wellbeing of drivers out and about. With generally record the road situations by using traffic cameras, it is achievable. what's more, critical to build up a strategy to naturally discover the oddities on the streets utilizing PC vision procedures. A traffic checking framework furnished with these calculations will achieve numerous advantages and comforts. On one hand, when the inconsistencies occur, a programmed framework can advise the traffic police promptly, to tackle the oddities on streets at the earliest opportunity. Then again, when arranging an outing, data about street conditions can give accommodation to the two drivers and travellers. Notwithstanding, it is an exceptionally moving assignment to plan a PC vision calculation to recognize abnormalities in street traffic. One principal reason is that the development examples of vehicles on streets are generally

exceptionally muddled, and distinctive irregular occasions may show complex practices. Simultaneously, the irregular occasion happens once in a while when contrasted with typical occasions. Accordingly, building up a productive and compelling clever calculation for programmed video abnormality location is a squeezing need. Numerous works of irregularity discovery in observation recordings must be applied to distinguish explicit strange occasions. For example, Mohammedi et al. build up a technique to distinguish human viciousness in recordings [4]. Additionally, the traffic finders just work in exceptionally restricted conditions [3]. With the improvement of traffic and video reconnaissance advancements, traffic the board frameworks dependent on video observation have gotten broadly utilized in rush hour gridlock the executives. In the canny preparing of traffic data, the discovery and acknowledgment of traffic inconsistencies, for example, securing, gridlock, car crashes, and unlawful driving have pulled in the consideration of numerous analysts because of its significance in rush hour gridlock the board. In any case, due to the intricacy of traffic camera feed, traffic reconnaissance video is helpless against outer factors, for example, light, climate, and checks. The constraints of existing picture preparing and investigation innovations

make the traffic boundaries (flow of the traffic, density of traffic, vehicle direction, speed, and so forth) in light of traffic video extremely dubious.

Looking the problem, we noticed it a bit difficult for manual surveillance. So, we came up with a proposal which is a CNN based surveillance system designed to learn for detecting the anomalies in dense traffic conditions, which handles of both the dynamic and static vehicles. Typically, the vehicles should continue to proceed onward the streets aside from certain ordinary conditions (e.g., hanging tight for traffic signals). Thusly, the static vehicles have higher likelihood for being anomalous occasions. By and large, the majority of the anomalous occasions in street deals will cause vehicle halting. For instance, vehicle slowing down, car crash or vehicle sticking. In the interim, the static vehicles can give us the exact area of the atypical occasions. To recognize the static vehicles from the moving traffic, we acquaint a static mode technique with get the running normal of the edge succession. Additionally, roused by the incredible accomplishment of deep learning in PC vision field, we send the deep learning-based strategy for the static vehicle identification. Additionally, we designed to further recognize the vehicle pictures of accidental pictures

2 Related Work

Vehicle monitoring, identification and tracking is the essential part in the street traffic scenario examination and assumes a significant part in many related applications, for example, driver help frameworks. Because of the extraordinary achievement of DL innovation, we have acquired a gigantic enhancement in image detection fields [2] [3] [4]. In this paper, we additionally experimenting the DL algorithms for detecting the vehicles and following. The detection of anomalies in both dynamic videos of the moving vehicles and static images of the traffic dataset have been concentrated in the previous years because of the expanding revenue in open security [3]. The customary techniques typically gain proficiency with the hand-created highlights to demonstrate the ordinary/anomalous occasion designs. As of late, deep learning innovation has been created for peculiarity detection as its accomplishment in the PC vision field [5].

In [6], the researchers present a generative adversarial network (GAN) based technique to identify the inconsistencies in pictures, utilizing just typical information to prepare the models. For observation recordings, there are a few endeavours to identify human savagery or strange occasions in group scenes. a deep peculiarity positioning system is used to foresee anomalies while testing on the recorded datasets. Since street condition assumes a significant part in our day-by-day life, recognizing peculiarities on streets has stood out from numerous researchers. For this undertaking, the main objective is to discover where and when the inconsistencies happen.

M. Schubert et al. [7] propose a mix of methods to foresee the event of street mishaps. The researchers

present a visual examination structure for the investigation of ordinary social models and the detection of peculiar occasions. Be that as it may, the new works are intended to distinguish a particular strange occasion, and this present reality irregularities on streets are convoluted and assorted. Subsequently, we plan a novel double model strategy to distinguish different street traffic odd occasions in genuine scenes, which can have wide use practically speaking. Few more pulled in numerous specialists to direct top to bottom examination on traffic irregularity detection.

In, [8] the authors proposed a vehicle vision monitoring calculation dependent on PC vision, which had high tallying exactness and improved the precision of traffic surveillance. Frejlichowski et al. [9] proposed another vehicle direction design acknowledgment calculation dependent on the Cam shift calculation, which can precisely dissect and recognize the illicit leaving or unlawful turning of vehicles. Yang [11] successfully distinguish traffic occurrences of turnpike scenes by utilizing fuzzy logic (FL) by consolidating FL and improved steady examination calculations in their proposed system. The system breaks down occasions by extricating traffic flow data and vehicle speed, yet the system detection has few restrictions because of the intricacy of conditions of a traffic. In [10] the researchers utilized the GPS information of a hire taxi, which also including direction and velocity, to recognize gridlock on metropolitan streets. Albeit the precision of GPS strategy is high, also it has the limitations of significant expense, which compelled its application. In-order to identify these problems, another calculation that incorporates more traffic boundaries is proposed in this paper. The proposed calculation cannot just recognize traffic anomalies all the more precisely and carefully yet additionally be viable in various circumstances.

3 System Model

3.1 VGG19 Architecture

VGGNet is a CNN(Convolutional Neural Network) architecture proposed by Karen Simonyan and Andrew Zisserman from Oxford university in 2014. This paper principally centers around the impact of the CNN profundity on its precision. The contribution to VGG based convNet is a 224*224 RGB picture. Pre-processing layer takes the RGB picture with pixel esteems in the scope of 0–255 and deducts the mean picture esteems which are determined over the whole ImageNet training set. The input pictures subsequent to pre-processing are gone through these weight layers. The training pictures are gone through a stack of convolution layers. There is an aggregate of 13 convolutional layers and 3 fully connected layers in VGG16 engineering. VGG has more modest channels (3*3) with more profundity as opposed to having enormous channels. It has wound up having a similar powerful open field as though you just have one 7 x 7 convolutional layers. Another model of VGGNet has

19 weight layers comprising of 16 convolutional layers with 3 fully connected layers and a similar 5 pooling layers. In the two varieties of VGGNet, there comprises of two Fully Connected layers with 4096 channels every which are trailed by another fully connected layer with 1000 channels to foresee 1000 marks. The last fullyconnected layer utilizes a softmax layer for order purposes.

The walk through of 19 layers as follows: The initial two layers are convolutional layers with 3*3 channels, and initial two layers utilize 64 channels that outcomes in 224*224*64 volume as same convolutions are utilized. The channels are consistently 3*3 with step of 1.

After this, pooling layer was utilized with max-pool of 2*2 size and step 2 which lessens tallness and width of a volume from 224*224*64 to 112*112*64.

This is trailed by 2 more convolution layers with 128 channels. This outcomes in the new component of 112*112*128.

Subsequent to pooling layer is utilized, volume is decreased to 56*56*128.

Two more convolution layers are added with 256 channels each followed by down inspecting layer that diminishes the size to 28*28*256.

Two more stack each with 3 convolution layer is isolated by a maximum pool layer.

Fig. 1. VGG19 Architecture

After the last pooling layer, 7*7*512 volume is straightened into Fully Connected (FC) layer with 4096 channels and softmax yield of 1000 classes.

3.2 ImageNet

ImageNet is a project which aims to provide a large image database for research purposes. It contains more than 14 million images which belong to more than 20,000 classes (or synsets). They also provide bounding box annotations for around 1 million images, which can be used in Object Localization tasks. It should be noted that they only provide urls of images and you need to download those images.

3.3 Traffic Net Dataset

The Traffic-Net dataset is a collection of traffic images that provide real-time monitoring, analytics, and alerts that will be used to provide a machine learning system with the data it needs to detect traffic conditions [12].

Using machine learning systems to accustom themselves to perception, understanding, and action in any environment is one of Deep Quest AI's goals [12].

4400 images are contained in the Traffic-Net dataset. The recent release includes the following classes:

- Accident
- Dense Traffic
- Fire
- Sparse Traffic

The images are divided into different categories with 900 images for training and 200 for testing. A new set of categories will be added in the future and the dataset will be improved [12].

Fig. 2. Sparse Traffic

Sparse Traffic: Sparse traffic dataset means less traffic or no traffic regions

Fig. 3. Fire Accident

Fire Accident: Fire accident dataset is the pictures of accidentally burned vehicles.

Fig. 4. Dense Traffic

Dense Traffic: Dense traffic dataset is a heavy traffic regions dataset.

Fig. 5. Accident dataset

Accident Dataset: Accident vehicles dataset in different regions.

3.4 Optimizer:

Adam is a deep learning algorithm that replaces stochastic gradient descent for the purpose of training models. The Adam algorithm combines the qualities of the AdaGrad and RMSProp algorithms to produce an optimization algorithm capable of handling sparse gradients on noisy problems. Adam can be configured quite simply, with default parameters that work out most of the time.

4. Proposed Method:

In this section, we are explaining the methodology followed to detect and classify the anomalies happened in various roads in various countries out of the varied traffic data-set. We are proposing an end-to-end deep learning architecture for traffic flow and anomalies detection and classification. We are experimenting using a VGG19 Convolutional Neural Network and ImageNet pretrained weights to discover the static vehicles on streets, as the abnormalities normally lead to vehicle halting. Typically, the majority of the inconsistencies on streets are unusual vehicle slowing down or vehicle crashes, which both make the vehicles stop in/alongside the street. In light of this perception, we acquaint a movement investigation strategy with discover the static vehicles which are met with accidents sometimes which get caught on fire on streets and further perceive the anomalous occasions dependent on that. The pipeline of peculiarity detection dependent on

the static vehicle is introduced in Figure 2, 3, 4 5. The dataset which is properly bifurcated the data into two separate folders i.e., one is test dataset and another one is train data set. In each of those training and test dataset is categorised the four different scenarios labelled namely “Accident, Sparse traffic, fire, and dense traffic”. Initially, we took the training dataset to train the model to classify its type of class. In order to train the model, we read all the images in each category mentioned above in the training directory using the OpenCV library. After reading the whole dataset, the images are resized to same height and width in pixels. Here in this experiment, we are considering it as 300 x 300(height, width). To ease the process the whole dataset of images which has 3 channels of every image is converted into a binary scale image which means converting the image channel from RGB to binary image which is giving the range from Zero to One. In same way the test directory images also read and resized all the data into same shape and converted the images into binary images for validation purpose. After aligning the image data the pretrained VGG19 model and ImageNet weights are taken to create a model which is shown below in Fig.6.

Instead of building the whole convolutional network from scratch and initializing weights to create a model we have chosen the pretrained model which is VGG19. Any model or classifier would at the initial stage be able to detect the slant lines whatever the class we expect to classify as the first step, according to the intuition of choosing the pre-trained model. The training of those in every possible opportunity to create a Neural network makes no sense. The project details that need to be trained will determine the abilities of the final layer of the network to identify classes. And also, there is one more advantage for using pretrained models which is pretrained models are always trained on big datasets or data which are not usually available to everyone. An example is ImageNet, which contains approximately 14 million images, 1.2 million of which are assigned to one of 1,000 categories. Thus, we would benefit tremendously from making use of these models” [2]. As with our previous experiment, instead of repeating the procedure for the first network and starting from scratch with random weight values, we can now use the saved weight values from the previous experiment as the initial weights for our new experiment. In this case, weights are initialized using pre-trained networks.

So, here we are taking the pretrained model and on top of it we are adding Flatten layer and dense layer. Using flatten layer function we flattened all the convoluted layers into a 1dimensional linear vector. Later drop out technique is used to dropout the nodes of the dense layer which is in our case having 1024 neurons in a dense layer out of it 30% of the neurons are dropped out at fully connected layer-1 and again out of 512 neurons of dense layer at fully connected layer 30% of them is dropped out.

```
6 vgg19.summary()
```

Model: "vgg19"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 300, 300, 3)]	0
block1_conv1 (Conv2D)	(None, 300, 300, 64)	1792
block1_conv2 (Conv2D)	(None, 300, 300, 64)	36928
block1_pool (MaxPooling2D)	(None, 150, 150, 64)	0
block2_conv1 (Conv2D)	(None, 150, 150, 128)	73856
block2_conv2 (Conv2D)	(None, 150, 150, 128)	147584
block2_pool (MaxPooling2D)	(None, 75, 75, 128)	0
block3_conv1 (Conv2D)	(None, 75, 75, 256)	295168
block3_conv2 (Conv2D)	(None, 75, 75, 256)	590080
block3_conv3 (Conv2D)	(None, 75, 75, 256)	590080
block3_conv4 (Conv2D)	(None, 75, 75, 256)	590080
block3_pool (MaxPooling2D)	(None, 37, 37, 256)	0
block4_conv1 (Conv2D)	(None, 37, 37, 512)	1180160
block4_conv2 (Conv2D)	(None, 37, 37, 512)	2359808
block4_conv3 (Conv2D)	(None, 37, 37, 512)	2359808
block4_conv4 (Conv2D)	(None, 37, 37, 512)	2359808
block4_pool (MaxPooling2D)	(None, 18, 18, 512)	0
block5_conv1 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv4 (Conv2D)	(None, 18, 18, 512)	2359808
block5_pool (MaxPooling2D)	(None, 9, 9, 512)	0
Total params: 20,024,384		
Trainable params: 20,024,384		
Non-trainable params: 0		

Fig. 6. VGG19 Pretrained Model

As the data we have chosen is having 4 different classes, so that the model has to classify the image of which class it belongs to. In order to do that the final layer has chosen a dense layer again and used SoftMax activation function which is used in the output layer of convolutional neural network for multi class classification problems. The model summary is shown in above figure-7.

To compile the model, we have chosen the Adam optimization algorithm for stochastic gradient decent the AdaGrad and RMSProp algorithms to provide an optimization algorithm which adjusts the weights and bias of the network to help learning and reduces the loss. We have then selected a loss function for our optimization function, which is categorical cross entropy, which is typically used in multi-class classification problems. The purpose of this measure is to compare the probability distributions of two events. After the loss function the whole agenda is to find out the accuracy and loss mitigation of the model we have chosen. To do that we have used model.fit() function for fitting the model in which we are separating the whole training dataset into 64 batches and 10 epochs, which means the whole dataset into 64 parts and passed through the network for 10 iterations.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 300, 300, 3)]	0
block1_conv1 (Conv2D)	(None, 300, 300, 64)	1792
block1_conv2 (Conv2D)	(None, 300, 300, 64)	36928
block1_pool (MaxPooling2D)	(None, 150, 150, 64)	0
block2_conv1 (Conv2D)	(None, 150, 150, 128)	73856
block2_conv2 (Conv2D)	(None, 150, 150, 128)	147584
block2_pool (MaxPooling2D)	(None, 75, 75, 128)	0
block3_conv1 (Conv2D)	(None, 75, 75, 256)	295168
block3_conv2 (Conv2D)	(None, 75, 75, 256)	590080
block3_conv3 (Conv2D)	(None, 75, 75, 256)	590080
block3_conv4 (Conv2D)	(None, 75, 75, 256)	590080
block3_pool (MaxPooling2D)	(None, 37, 37, 256)	0
block4_conv1 (Conv2D)	(None, 37, 37, 512)	1180160
block4_conv2 (Conv2D)	(None, 37, 37, 512)	2359808
block4_conv3 (Conv2D)	(None, 37, 37, 512)	2359808
block4_conv4 (Conv2D)	(None, 37, 37, 512)	2359808
block4_pool (MaxPooling2D)	(None, 18, 18, 512)	0
block5_conv1 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block5_conv4 (Conv2D)	(None, 18, 18, 512)	2359808
block5_pool (MaxPooling2D)	(None, 9, 9, 512)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 1024)	42468352
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 4)	2052
Total params: 63,019,588		
Trainable params: 42,995,204		
Non-trainable params: 20,024,384		

Fig. 7. VGG19 Model summary

For each iteration the whole batch of images passes through the network and the optimization function adjusts the weights of the matrix and the loss function algorithm reduces the loss and checks with validation data. The Fitting model is shown in below figure-8.

5. Results

We successfully implemented the VGG19 model to detect the anomalies of the road and achieved model accuracy of 95.62% and model loss of 13% in the eighth iteration and


```
Epoch 1/10  
57/57 [=====] - 1153s 28s/step - loss: 5.3036 - accuracy: 0.4449 - val_loss: 0.4963 - val_accuracy: 0.8225  
  
Epoch 00001: val_accuracy improved from -inf to 0.82250, saving model to ./best_weights.hdf5  
Epoch 2/10  
57/57 [=====] - 1477s 26s/step - loss: 0.4875 - accuracy: 0.8252 - val_loss: 0.4891 - val_accuracy: 0.8413  
  
Epoch 00002: val_accuracy improved from 0.82250 to 0.84125, saving model to ./best_weights.hdf5  
Epoch 3/10  
57/57 [=====] - 1265s 22s/step - loss: 0.3004 - accuracy: 0.8949 - val_loss: 0.3777 - val_accuracy: 0.8662  
  
Epoch 00003: val_accuracy improved from 0.84125 to 0.86625, saving model to ./best_weights.hdf5  
Epoch 4/10  
57/57 [=====] - 1132s 20s/step - loss: 0.2165 - accuracy: 0.9224 - val_loss: 0.4482 - val_accuracy: 0.8358  
  
Epoch 00004: val_accuracy did not improve from 0.86625  
Epoch 5/10  
57/57 [=====] - 1345s 24s/step - loss: 0.2586 - accuracy: 0.9002 - val_loss: 0.4322 - val_accuracy: 0.8325  
  
Epoch 00005: val_accuracy did not improve from 0.86625  
Epoch 6/10  
57/57 [=====] - 1226s 22s/step - loss: 0.1933 - accuracy: 0.9232 - val_loss: 0.4689 - val_accuracy: 0.8525  
  
Epoch 00006: val_accuracy did not improve from 0.86625  
Epoch 7/10  
57/57 [=====] - 1164s 20s/step - loss: 0.1796 - accuracy: 0.9318 - val_loss: 0.5399 - val_accuracy: 0.8375  
  
Epoch 00007: val_accuracy did not improve from 0.86625  
Epoch 8/10  
57/57 [=====] - 1118s 20s/step - loss: 0.1349 - accuracy: 0.9563 - val_loss: 0.5748 - val_accuracy: 0.8075  
  
Epoch 00008: val_accuracy did not improve from 0.86625  
Epoch 9/10  
57/57 [=====] - 1106s 19s/step - loss: 0.1896 - accuracy: 0.9273 - val_loss: 0.4999 - val_accuracy: 0.8487  
  
Epoch 00009: val_accuracy did not improve from 0.86625  
Epoch 10/10  
57/57 [=====] - 1124s 20s/step - loss: 0.2210 - accuracy: 0.9183 - val_loss: 0.4543 - val_accuracy: 0.8525  
  
Epoch 00010: val_accuracy did not improve from 0.86625
```

Fig.8. Trained model **Fig. 9.** Model Accuracy and Loss graph

we have achieved the validation accuracy of 86% and accuracy loss 45%.

6. Conclusion

In this paper we present a VGG19 CNN model-based traffic anomalies detection in urban traffic environments and achieved the highest accuracy of 95.6%. As this model is very feasible and accurate, when implements it in cloud environments and used for traffic surveillance purposes there will be a great chance of communicating the helpline at desperate situations and also helps to decreases the accident probability.

7. Future Scope

This research work have a great scope towards self-driving cars, video surveillance implementation and Accident detection system development in loop line areas, accident prone zone areas and Ghat roads. This research work is continued for further enhancements to implement in embedded hardware and cloud environments and optimize the code complexity to decrease the computing time.

References

- [1] S. Kamijo, Y. Matsushita, K. Ikeuchi, M. Sakauchi. IEEE Transactions on Intelligent Transportation Systems, **1**, 108 (2000).
- [2] K. He, G. Gkioxari, P. Dollar, R. Girshick. *Mask r-cnn*, IEEE International Conference on Computer Vision, 2980, (2017)
- [3] S. Ren, K. He, R. Girshick, J. Sun. IEEE transactions on pattern analysis and machine.
- [4] H. Nam, B. Han., *Learning multi-domain convolutional neural networks for visual tracking*. IEEE Conference on Computer Vision and Pattern Recognition, 4293, (2016)
- [5] M. Sabokrou, M. Fayyaz, M. Fathy, R. Klette. IEEE Transactions on Image Processing, **26**, 1992, (2017)
- [6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. ,Advances in Neural Information Processing Systems, 2672, (2014)
- [7] M. Deublein, M. Schubert, B. T. Adey, J. Kohler, M. H. "Faber, Accident Analysis & Prevention, **51**,274(2013)
- [8] N. Seenoung, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, N. Ohnishi, *A computer vision based vehicle detection and counting system*, 8th International Conference on Knowledge and Smart Technology (KST), (Chiang Mai, Thailand, 224, 2016)
- [9] Nowosielski, A., Frejlichowski, D., Forczmański, P., Gościńska, K., Hofman, R., Automatic Analysis of Vehicle Trajectory Applied to Visual Surveillance (2016).

- [10] “Li Ning, Li Gang, R.C Xie, Yuan Hang.,Computer Simulation., **32**,113 (2015)
- [11] “Yang Zhiyong, Ma Hongwei, Chen Xiaoping. Freeway Incident Detection Algorithm Based on Fuzzy Logic. Journal Of Chong Qing University(Natural Science)., 32(6):1247- 1251(2013)
- [12] trafficnet, [https://github.com/ Olafenwa Moses/ TrafficNet/ releases/tag/1.0](https://github.com/OlafenwaMoses/TrafficNet/releases/tag/1.0)”.
- [13] “[https://machinelearningmastery.com/adamoptimization-algorithm-for- deeplearning/#:~: text= Adam%20is%20a%20replacement%20optimization, sparse%20gradients%20on%20noisy%20problems.](https://machinelearningmastery.com/adam-optimization-algorithm-for-deeplearning/#:~:text=Adam%20is%20a%20replacement%20optimization,sparse%20gradients%20on%20noisy%20problems.)”
- [14] “<https://towardsdatascience.com/how-dopretrained-models-work-11fe2f64eaa2>”.
- [15] “[https://peltarion.com/knowledgecenter/documentati on/modeling-view/build-an-aimodel/loss- functions/categorical-crossentropy](https://peltarion.com/knowledgecenter/documentation/modeling-view/build-an-aimodel/loss-functions/categorical-crossentropy)”.