Video-based Car Make, Model and Year Recognition

Diana Atef George[1], Omar M. Shehata[1], Hossam E. Abd El Munim[2] and Sherif Hammad[1]
[1]*Mechatronics Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt*
[2]*Computer and Systems Engineering Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt*

Keywords: Fine-grained Car Recognition, CNN, Tracking, Detecting.

Abstract: Fine-grained car recognition requires extracting discriminating features and certain car parts which can be used to distinguish between similar cars. This paper represents a full system for car make, model and year recognition in videos. We followed a multi-step approach for automatically detecting, tracking and recognizing them using deep Convolutional Neural Network (CNN). We also focused on the recognition stage where we managed to compare 4 state-of-the art Convolution Neural Networks and adapted them for extracting those features. Moreover, we modified the InceptionResnetv2 network and our results show our success as we managed to elevate the Top 1 accuracy to 0.8617 and Top 5 accuracy to 0.9751.

1 INTRODUCTION

Machine learning has been widely used in many applications including medical diagnosis, treatment and prognosis. Moreover, it is used in natural language processing, speech recognition, recommendation systems, facial recognition and fraud detection. Fortunately, with the advancements in deep learning it has been possible to mimic the human brain and even surpassing the human accuracy. However, it is still extremely challenging when it comes to fine-grained classification. As, it is somehow difficult for a machine to learn discriminative features and distinguish between similar classes.Therefore, there is always an attempt to use deep learning in such applications for automatically learning those specific features. Fine-grained classification include discriminating birds(Gavali and Banu, 2020),flowers(Nguyen et al., 2016) ,cars(Liu and Wang, 2017)and many more.

Certainly, car classification in videos is considered a tremendously significant task and is required for critical applications such as intelligent transportation systems for traffic and car crash analysis. However, the accuracy of the models in such applications is still not high enough to be deployed in the real world. As, it includes many challenges including tiny changes between different car models, changes in illumination conditions and occlusions which make the task even harder. Although it is a complex task, it is made up of mainly three stages: car detection, tracking and recognition.

For object detection there are many algorithms that could be used such as a sliding window but it is slow and not used in real time detection. To speed up the process R-CNN (Girshick et al., 2015), faster RCNN (Ren et al., 2015) and single shot MultiBox(Liu et al., 2016) are used instead. Yolov3 (Redmon and Farhadi, 2018) is considered the best when it comes to real-time object detection. As proved by Bilel, et al(Benjdira et al., 2019) YOLOv3 is better than Faster RNN and only took on average 0.057ms to process an image while Faster RNN took 1.39s.

For car classification, Fomin, et al (Fomin et al., 2020) combined the car classification algorithm with car position as viewed from the camera achieving a precision of 92 % on CompCars dataset (Yang et al., 2015) using InceptionResnetv3, 98 % for car detection using YOLOv3 and 96% for car direction classification. Hu, Qichang, et al (Hu et al., 2017) used spatially weighted pooling and achieved an accuracy of 93.1% using ResNet101-swp on Stanford cars dataset (Krause et al., 2013) for car model detection , an accuracy of 97.6% on car model and an accuracy of 99.3% on car make of CompCars dataset. Dehghan, Afshin, et al (Dehghan et al., 2017) collected data and preprocessed them using Sighthound Cloud API for aligning labeled car to the center. They trained 2 Deep Neural Networks , one for detecting car make and model and the other for detecting color. They achieved an accuracy of Top 1 93.6% on Stanford car dataset and a Top 1 95.88% on CompCar dataset. Liu, et al (Liu

George, D., Shehata, O., El Munim, H. and Hammad, S.
Video-based Car Make, Model and Year Recognition.
DOI: 10.5220/0010649900003061
In *Proceedings of the 2nd International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2021)*, pages 86-91
ISBN: 978-989-758-537-1

and Wang, 2017) tried out different CNN models to find the best architecture and achieved Top 1 accuracy of 80% and Top 5 accuracy of 95.1% on Stanford cars dataset using GoogLeNet. Alsahafi, Yousef, et al (Alsahafi et al., 2019) used SSD-CNN architecture. A video of 25 fps was input to SSD300 for object detection then they cropped the images and passed them over to ResNet152 CNN for classification achieving 76.18% accuracy as the SSD missed some of the bounding boxes and some of the cars were misclassified. For car model recognition of 107 classes Gao, Yongbin, et al (Gao and Lee, 2015) used frame difference for localizing moving cars in videos and the output binary image is then passed to a symmetry filter to detect frontal view of the car which is then passed to a CNN. They achieved an accuracy of 88.4% for car recognition and 100% frontal view accuracy.

In this research paper we introduce a whole system for vehicle recognition in videos. We also compare state-of-the-art CNNs to find the best architecture. Moreover, we modified the InceptionResnetv2 network and succeeded in achieving a higher accuracy. The rest of the paper is divided as follows: First, we provide an overall and detailed system description. Then, we describe how we adapted and modified neural network models to solve such problem, dataset and hardware used. Finally, there is a section for all the experimental results, discussion, conclusion and future work.

2 SYSTEM DESCRIPTION

The system is composed of 3 modules: Car detection,Tracking and Recognition modules as shown in Fig. 1.

First, a video frame is passed to the car detection module where all the cars that are in the frame are detected and the bounding boxes are then passed to the car tracking module.

The Tracking module is responsible for deciding, for each car detected, whether to retrieve the car model stored as it was previously classified or pass it to the car recognition module to classify, save the result and keep tracking it for the subsequent frames until it disappears.

The Recognition module uses CNN to predict the cars' make, model and year.

2.1 Detection Module

We used YOLOv3 on a video stream of 25 FPS to extract the exact locations of the cars in video frames and send their bounding boxes to the car tracking module. Confidence of less than 0.85 was rejected. YOLOv3 detection rate depends on the hardware it is running on. On NVIDIA GTX 1660 TI YOLOv3 processed the images at around 15 FPS but on a Pascal Titan X it processed images at 30 FPS.

2.2 Tracking Module

We first need to calculate the Maximum Distance which will be used in tracking the cars so as to decide whether to consider the car the same as that previously recognized in the previous frame or not using equation 1 where the height and width of detection from the previous frame are used for calculation.

$$Maximum\ Distance = 0.4 * \sqrt{h^2 + w^2} \quad (1)$$

where:

h = height of detection from the previous frame
w = width of detection from the previous frame

The Tracking module calculates the euclidean distance between the centroid of the bounding boxes detected by Yolov3 and the previously stored ones to decide whether the car was previously recognized or not. If the euclidean distance is less than the Maximum Distance calculated, then it is the same car; thus retrieving previously stored classification. if not, then the module uses the bounding box to crop the car and resize it to 300×300 pixels for the recognition module to be able to classify the car and then store the recognition output. The module also checks if any car has disappeared by keeping track of unassociated stored centroid.If it was kept unassociated for more than 15 frames then it should be removed from the stored centroids as the object has already disappeared. We modified the simple tracking algorithm developed by (Rosebrock, 2018) to adapt it for tracking only the cars class and recognize their make, model and year as well. Tracking Algorithm is as shown in Algorithm 1 starting with the input frame from the video as an input.

2.3 Recognition Module

In car recognition we initialized our networks using ImageNet weights as Transfer learning is vital to speed up the learning process where the earlier layers extract almost the same features and then we fully trained them using Stanford Cars dataset. We modified state-of-the-art networks and added a softmax layer with 196 neurons for classifying multi-classes present in the dataset. Furthermore, we modified the

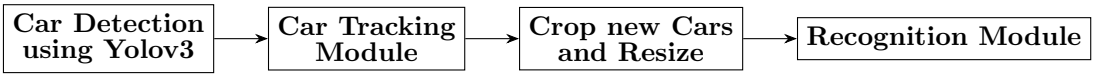

Figure 1: System Block diagram which is composed of the Detection module for creating the bounding boxes, Tracking module for tracking cars and the recognition module to use CNN for recognizing new unpredicted cars.

Algorithm 1: Cars Tracking Algorithm.

Input: Center of Bounding Boxes detected by Yolov3
Output: Cars Recognized in Frame
Bounding Boxes detected: Calculate the Euclidean Distance
if *Distance* $<$ *Max Distance* **then**
    Associate to previously recognized car
else
    Crop and Send for Recognition
    Strore Centroid and car recognized
Check Stored Centroid
if *left unassociated* **then**
    Check Disappeared Counter
    if *Disappeared Counter* $<$ *15* **then**
        Increment Disappeared Counter
    else
        Remove Centroid
goto Bounding Boxes detected

Inceptionresnet network by adding a dense layer of 1024 neurons before the softmax layer for improving the accuracy. We modified and tested the following :

- InceptionResNetV2(Szegedy et al., 2016a) is made up of 164 layers and is a combination of inception architecture along with residual connections. Modification is as shown in Fig. 2 with the dense layer added before the softmax.
- Inceptionv3(Szegedy et al., 2016b) is made up of 42 layers deep which aims to reduce the number of parameters for reducing the probability of overfitting while not affecting the efficiency of the network. We modified it by adding a softmax layer of 196 neurons to classify 196 car classes.
- MobileNetV2(Sandler et al., 2018) was also developed in an attempt to reduce the number of parameters without affecting the accuracy of the network. We modified it by adding a softmax layer of 196 neurons to classify 196 car classes.
- Resnet50(He et al., 2016) was mainly targeting the vanishing gradient problem which arises in very deep neural networks making them difficult to train so they stacked residual blocks together instead of using a plain network. We modified it by adding a softmax layer of 196 neurons to classify 196 car classes.

3 DATASET, HARDWARE USED AND IMAGE PREPROCESSING

We used Stanford car dataset (Krause et al., 2013) which was released in 2013 and is made up of 196 classes containing a collection of 16,185 car images taken from different angles, resolution and illumination defining their make, model and year as well as their bounding boxes. We used 8144 images for training, 4020 for validation and 4021 for testing.

NVIDIA TESLA P100 GPU was used due to its high speed in neural network training.

The bounding boxes provided were used to crop the images and then we resized them to 300×300 pixels for training and testing. We sheared, zoomed, rotated and flipped the cars horizontally as data Augmentation is required to avoid overfitting.Training data was normalized to have a mean of zero and a variance of one to speed up the learning process.

4 EXPERIMENTAL RESULTS

4.1 Training and Testing Results

First, We conducted 5 experiments to find out the best architecture and to test our modified Inceptionresnetv2 with the dense layer added. We set the learning rate to 0.0001, used adam optimizer and early stopping to avoid overfitting. As shown in Table 1, Table 2, Fig. 3 and Fig. 4 our modified Inceptionresnetv2 can extract the discriminating features in cars achieving a higher accuracy without overfitting the training dataset due to the presence of a dense layer and using L2 regularization.

Then, we performed experiments with state-of-the-art InceptionResnet and our modified Inception-Resnet trained for 50 epochs. Our modified network outperforms Top 1 accuracy achieved by state-of-the-art InceptionResnet and increased by 2 % . We also outperform Liu, et al's Googlenet(Liu and Wang, 2017)by 6 % as shown in Table 3 and Fig. 6.

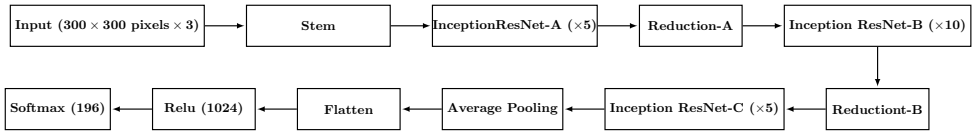

Figure 2: Our Modified InceptionResnetv2 with a fully connected layer of 1024 neurons added before the softmax layer of 196 neurons to classify the 196 car classes present in the dataset.

Table 1: Make, Model & Year Top 1 Accuracy using different CNN showing our modified Inceptionresnetv2 outperforming the other networks achieving higher validation accuracy.

Model	Epochs	Train. Acc.	Valid. Acc.
1.InceptionResnetv2	14	0.8001	0.7423
2.Inceptionv3	18	0.8152	0.7229
3.MobileNetv2	10	0.5156	0.3694
4.Resnet50	19	0.6787	0.4898
5.Modified-InceptionResnetv2	18	0.7744	**0.7652**

Table 2: Make, Model & Year Top 5 Accuracy using different CNN showing our modified Inceptionresnetv2 outperforming the other networks achieving higher validation accuracy.

Model	Epochs	Train. Acc.	Valid. Acc.
1.InceptionResnetv2	14	0.931	0.946
2.Inceptionv3	13	0.944	0.929
3.MobileNetv2	10	0.887	0.839
4.Resnet50	19	0.876	0.716
5.Modified-InceptionResnetv2	18	0.943	**0.952**

Table 3: Comparison between Top 1 and Top 5 Testing accuracy (Acc.) of different CNN models, with our model trained for 50 epochs.

Model	Top 1 Acc.	Top 5 Acc.
1.Modified-InceptionResnetv2	**0.8617**	**0.9751**
2.InceptionResnet	0.8436	0.9744
3.Liu GoogleNet(Liu and Wang, 2017)	0.8000	0.9510

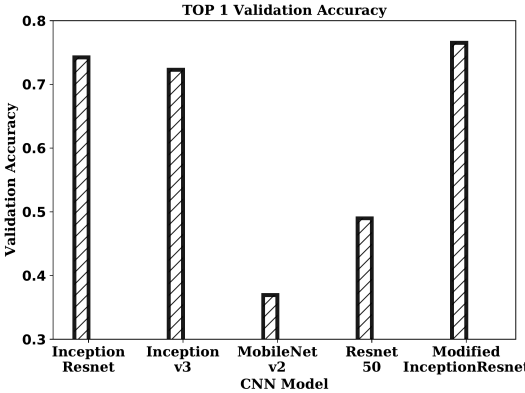

Figure 3: Top 1 Validation Accuracy Comparison between different Neural Networks with Adam optimizer and Early stopping.

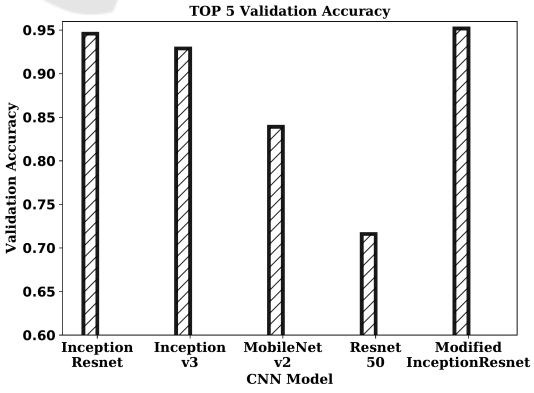

Figure 4: Top 5 Validation Accuracy Comparison between different neural network architectures used.

(a) Consecutive Frames from BMW car video

(b) Consecutive Frames from Audi car video

Figure 7: Consecutive Frames from different car videos using our Tracking and Modified-InceptionResnetv2 recognition model.

Figure 5: A sample of Cars Make, Model and Year correctly Recognized using our modified InceptionResnet.

4.2 Testing on Images and Videos

We tested our model using car images with different poses and illumination as shown in Fig. 5.

Also, we tested it on different videos as shown in Fig. 7. To evaluate our performance on the videos, we first created the ground truth of the cars' bounding boxes using the MATLAB Ground Truth Labeler App. This was used to output the coordinates of the bounding boxes around the cars which were then used to be compared with that predicted by our model. Average IoU was used as our evaluation metric which is

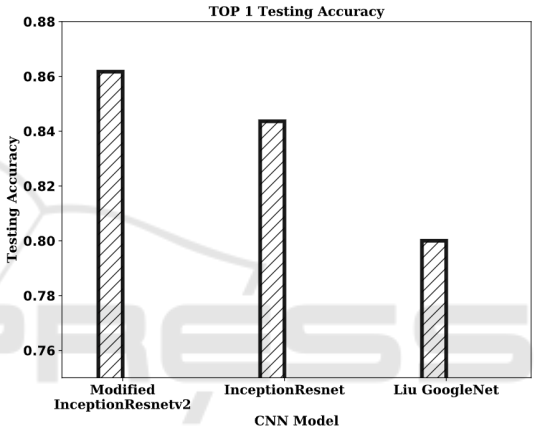

Figure 6: Top 1 Testing Accuracy Comparison between different neural network architectures with our model outperforming other networks.

the area of intersection over union between the ground truth and the model prediction. Our model managed to achieve an average IoU of 0.7 which is considered to be good enough to predict the approximate position of the cars in the videos.

5 CONCLUSION AND FUTURE WORK

Fine-grained car recognition in video is a complex task but can be subdivided into 3 tasks: Detection, Tracking and Recognition. In our paper we focused on comparing and representing the best CNNs which can be used by others to solve such an extremely important problem and with our network architecture modification we managed to significantly elevate the accuracy as represented in the paper.

For recognition, InceptionResnetv2 and Inceptionv3 are better than Resnet50, MobileNetv2 neu-

ral networks in fine-grained classification applications as they contain more number of layers with more number of trainable parameters which the model can use for learning discriminative features allowing it to distinguish between similar classes. Also, adding a dense layer to the InceptionResnetv2 allows for more features to be extracted while using L2 regularization to prevent overfitting the training dataset.

Our future work will include training and evaluating our models using (Buzzelli and Segantin, 2021) dataset to provide us with a larger and a broader set of annotations than the Stanford car dataset as it includes Type-level annotations for all CompCars models.

REFERENCES

Alsahafi, Y., Lemmond, D., Ventura, J., and Boult, T. (2019). Carvideos: A novel dataset for fine-grained car classification in videos. In *16th International Conference on Information Technology-New Generations (ITNG 2019)*, pages 457–464. Springer.

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., and Ouni, K. (2019). Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pages 1–6. IEEE.

Buzzelli, M. and Segantin, L. (2021). Revisiting the compcars dataset for hierarchical car classification: New annotations, experiments, and results. *Sensors*, 21(2).

Dehghan, A., Masood, S. Z., Shu, G., Ortiz, E., et al. (2017). View independent vehicle make, model and color recognition using convolutional neural network. *arXiv preprint arXiv:1702.01721*.

Fomin, I., Nenahov, I., and Bakhshiev, A. (2020). Hierarchical system for car make and model recognition on image using neural networks. In *2020 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pages 1–6. IEEE.

Gao, Y. and Lee, H. J. (2015). Vehicle make recognition based on convolutional neural network. In *2015 2nd International Conference on Information Science and Security (ICISS)*, pages 1–4. IEEE.

Gavali, P. and Banu, J. S. (2020). Bird species identification using deep learning on gpu platform. In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pages 1–6.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hu, Q., Wang, H., Li, T., and Shen, C. (2017). Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.

Liu, D. and Wang, Y. (2017). Monza: image classification of vehicle make and model using convolutional neural networks and transfer learning.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Nguyen, N., Le, V., Le, T., Hai, V., Pantuwong, N., and Yagi, Y. (2016). Flower species identification using deep convolutional neural networks.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Rosebrock, A. (2018). Simple object tracking with opencv. In *Online*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016a). Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016b). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Yang, L., Luo, P., Change Loy, C., and Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3973–3981.