ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about

Muhammad Awais[1]
Muhammad Bilal Qureshi[2]
Naila Hamid[3]
Asad Hussain Shah[4]

Segmentation For Object-Based Image Analysis (OBIA) Using Tensorflow Framework

Muhammad Awais
MS Computer Science, Department of Computer Science and IT, Superior University Lahore, 54000, Pakistan, Superior University (Sargodha Campus). awaixam1@gmail.com

Muhammad Bilal Qureshi
Department of Computer Science and IT, Superior University Lahore, 54000, Pakistan, Superior University(Sargodha Campus). bilalshah1728@gmail.com

Naila Hamid
UMT, Sialkot.
naila.hamid@skt.umt.edu.pk

Asad Hussain Shah
M Phil in Computer Science from Riphah International University.
asadhussainshah859@gmail.com

Abstract

Deep learning is the ultimate breakthrough of artificial intelligence and it will change the world dramatically in this century. Various type of deep neural networks has been used to resolve challenging computer vision problems such as detection, localization, recognition and segmentation of objects in the wild. Semantic segmentation to separate a portrait from the video background. Semantical Segmentation. This process essentially closes the image bits based on an object class to vacuum them together. In this paper we differentiate four separate deep learning models that we trained to provide real -time webcam video segmentation of portrait images, and analyze performance on the respective phases. They employ two distinct deep learning architectures and they employ two different datasets. Their data sets more than 30,000 human portrait images each. TensorFlow and Keras –° This is how we train our models (approach 2). Architecture 1 takes RGB images of 256×256 as input with around 12–14 FPS at runtime compared to Architecture 2 which takes RGB images of128 ×128 with runtimes at around15-18FPS. In our work, we outlined a principled method that earned greater accuracy and efficiency than any method described. Tags: portrait segmentation, semantic segmentation, TensorFlow in Flutter is the process of integrating TensorFlow, an open-source machine learning framework widely used for building machine learning applications, into a Flutter app. TensorFlow gives powerful library for model building and deployment for machine learning application, while flutter is a library for creating good-looking, well-designed interfaces.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

INTRODUCTION

Computer vision [1] is the science of letting just computers understand or manipulate image and videos. It could be employed in a whole range of applications, from driverless vehicles to factory inspection to augmented reality. In computer vision, deep learning for images and videos can be categorized into: classification, detection, segmentation and generation supplies categories. Deep learning is a group of methods derived from the domain of the Artificial Neural Network (ANN) which is part of the larger machine-learning community. A group of algorithms which imitate the human brain are called Artificial Neural Networks (ANN) [4]. The analogy with biological neurons comes from the facts that artificial neurons try to simulate a biological neuron behavior using numerical operations with real values. These artificial neurons are called perceptron's. A Slave Neuron/Perceptron takes multiple inputs, computes the weighted average of all these outputs and then passes this as a function to generate an output. A perceptron; its weight is learned when it is trained, based on the samples used for training. It all started with one neural network convolutional neural network (CNN) in the time of deep learning happened in computer vision. CNN [6] is one of the most used deep learning architectures to cope with the computer vision tasks as image segmentation. Monotonically evolving, CNNs have also been expanded in the area of video understanding (Wang et al. [6]). These are the filters, i.e. kernels, of CNNs Convolution layers are used for performing the convolution operation to extract significant features from the input. CNNs have weights, biases and outputs through a Nonlinear Activation [1]. NNs learn weights from inputs and through connected neurons in the subsequent layers Note that neurons in any given layer do not connect to each other. This is due to the fact that if images were regular neural networks, the size of the network would be super large due to having too many neurons, which would lead to overfitting. Visualize an image as a 3D volume (height, width and depth). On R B G, a depth is an image plane. CNN neurons are volumetric, as volumetric features are utilized for any deep learning framework for computer vision, CNN is one of its vital constituents. In the field of computer vision, semantic image segmentation is a very active research problem that can be tackled using CNNs with the objectives of achieving high accuracy and efficiency. As explained earlier, semantic segmentation [2] is merely pixel-wise classification. Semantic Segmentation gives a class to each pixel in the image, so pixels belonging to the same class get "label" or semantic meaning. Segmentation tasks are expensive to label.

The annotators who are software professionals, quite patient as well as accurate of their tasks Requires the semantic segmentation dataset [4]. Pixel-level accuracy labeling is widely regarded as the most tedious process of all annotation types. Hence it is challenging as the datasets of semantic segmentation are scarce and of less number of images. TensorFlow is a powerful open-source library for end-to-end machine learning [5]. Keras will still be compatible with its rich ecosystem of tools, libraries and community resources that inspire researchers to push the state-of-the-art in machine learning, and developers to create and deploy machine learning-powered applications. You also have support for a GPU for some NVIDIA cards, when it uses the corresponding version of CUDA toolkit [7]. An open source machine-learning library built by Google, TensorFlow has a robust and visual community for machine learning with strong support down to deep learning — the very same operation of pliable numerical computation that gets use in many other scientific areas. TensorFlow itself is written in C++, and it has a Python programming interface. The TensorFlow library uses Keras as a high-level API associated with the library [3]. It is commonly called tf. keras. Keras is one of the popular Deep Learning Library because of its tight integration with TensorFlow which has a more resilient reputation in Production. With TensorFlow 2.0 a lot has changed (the default execution mode was changed to eager execution and some APIs were completely removed [4]. for the framework as well. But Keras is not some high-level wrapper over TensorFlow, CNTK. It's an API for model creation and learning which is used to define machine learning models. TensorFlow eager execution is an imperative programming environment that evaluates operations immediately, without building graphs: operation results return concrete values instead of building a computation graph to execute later 5. Eager execution allows better-engineered software which is what TensorFlow 2.0 is all about. The paper uses TensorFlow2. 0 for the Keras API dimensionality. Primary language is Python 3. We propose four real-time deep learning based models for webcam portrait segmentation. This uses two architectures as well as datasets. Various architectures use convolutional neural networks.

Torchvision is an important part of the PyTorch ecosystem and is primarily computer vision utility library that implements image classification, detection, segmentation, image transformations. In-depth explanation: Built on top of PyTorch, Torchvision is a Python package that provides access to many popular datasets and models to expedite the processes of data preprocessing and training deep learning models. One of the key benefits of using this library is that many of the state-of-the-art models are provided, with architectures such as Faster R-CNN, YOLO, and RetinaNet also supported directly as well as multiple ResNet architectures for feature extraction and transfer learning, as the models can be used on a variety of platforms and do not require extensive computational power to run. Besides, Torchvision is bundled with the transforms module as well, making it easy to preprocess image data to get it ready for a deep learning model via resizing, normalization, and conversion to tensors. The datasets module makes loading and processing scalable datasets like ImageNet, COCO, and CIFAR-10 easier, and homogenizes the data pipeline for training and evaluation. For object detection tasks, Torchvision's pre-trained Faster R-CNN with Feature Pyramid Network (FPN) enables multi-scale detection, thereby enhancing accuracy for a variety of object sizes. The library also offers support for custom dataset integration, making it possible for researchers to fine-tune models on domain-specific datasets. Torchvision plays a pivotal role in academic research, industrial applications, and real-world deployments, beyond just autonomous systems, medical imaging, and surveillance, and its modularized design minimizes the complexities associated with executing the aforementioned purposes with low-complexity deep learning based end-model.

LITERATURE REVIEW

The Object-Based Image Analysis (OBIA) approach has gained popularity in the fields of remote sensing and computer vision due to its ability to classify and interpret high-resolution imagery at finer spatial resolutions accurately. OBIA is different from the pixel-based because OBIA arranges the pixels into meaningful objects considering the spectral and spatial information. Table 1 outlines the usage of deep learning frameworks, e.g. TensorFlow, in combination with OBIA in more

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

recent years, greatly contributing to more accurate (machine) segmentation due to more automation capabilities. The open-source library TensorFlow delivers powerful resources for building convolutional neural networks (CNNs) and other deep learning frameworks used to refine object segmentation in an image using feature extraction and pattern recognition. Different models, such as the HOG detector, R-CNN and Fully Convolutional Networks (FCNs), Fast RCNN, SPPNet, Faster RCNN available in TensorFlow, have confirmed the increased efficacy of these deep learning models in the segmentation of OBIA-based algorithms by learning effectively to discriminate between complicated land cover classes. Moreover, hybrid methods that integrate conventional OBIA strategies with deep learning approaches have been investigated to enhance segmentation regions and classification outcomes. There are still challenges to overcome such as the requirement of large datasets of labeled data, large computational costs, and the challenges of hyper parameter tuning model architectures for specific use-cases. Nevertheless, the advancements in software tools like TensorFlow and hardware resources such as GPUs, complementing the ongoing developments in data augmentation techniques, would be sparking innovation in OBIA segmentation to make it a competitive area for future research and to be applied in real-world areas like environmental monitoring, agricultural, and urban planning.

DIFFERENT METHODS FOR OBJECT DETECTION

HOG DETECTOR

The Histogram of Oriented Gradients (HOG) feature was first introduced by N. Dalal and B. Triggs in its initial form in 2005. HOG can be interpreted as a significant advancement over the scale-invariant feature transform and shape contexts of the time. To trade-off between feature invariance (e.g. translation, scale, illumination, etc) and linearity (i.e. on classifying different objects categories), the HOG descriptor was proposed to be computed over a very dense grid of uniformly spaced cells focused to use overlapping local contrast normalization on "blocks" for performance improvement. Though it can be used for detection for many different classes of objects, HOG was primarily motivated by the task of pedestrian detection. Size invariance is accomplished by the HOG detector that rescales the input image for multiple times, keeping the size of a detection window unchanged. The HOG detector has been a key building block of many object detectors as well as a wide range of computer vision applications for many years[26].

MILESTONES: CNN BASED TWO-STAGE DETECTORS

Human-made features have been saturating for a long time and object detection did not advance significantly after 2010. As evidenced by (R. Girshick): … progress was slow through 2010-2012 with small improvements gained by constructing ensemble systems and using minor variants of successful methods. The renaissance of the convolutional neural networks happened in 2012. As a deep conv net can learn to produce high-level feature representations of an image, it's a natural question to see if we can apply it to object detection. R. Girshick et al. ensured the move of object detection beyond the deadlocks by proposing the Regions with CNN features (RCNN) in 2014. From that point onward, object detection began its rapid evolution to where the state of the art is today. Thus, in the era of deep learning, object detection is divided into two detections: in "two-stage detection", detection is regarded as a "coarse-to-fine" process, but in "one-stage detection" it is regarded as "one-step completion".

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

RCNN

The basic idea behind RCNN is quite simple: It begins with a set of object proposals (object candidate boxes) are extracted by selective search [42]. Each proposal is then resized to a fixed size image and the CNN model pre-trained on the ImageNet dataset (e.g., AlexNet [40]) is applied to extract features. Finally, linear SVM classifiers are applied to predict the presence of an object in each region and recognize object categories. 4 RCNN achieves a remarkable performance gain on VOC07, improving the state-of-the-art mean Average Precision (mAP) significantly from 33.7% (DPMv5) to 58.5%. Despite RCNN has advances, its shortcomings are equally apparent, where the same features are extracted over multiple overlapping proposals (over 2000 boxes from one image), which leads to an extremely slow detection rate (14s per image with GPU). In the same year, SPPNet proposed and solved this issue [26].

SPPNET

In 2014, K. He et al. de-synthesized a Spatial pyramid pooling networks(SPP- net). Note uses a fixed-size input in previous CNN models, such as a 224x224 image for AlexNet. SPPNet presents the SPP layer, a new neural network layer that is capable of allowing a CNN to create a fixed length representation regardless of the image / region of interest size, and without the need to scale the image. For example, with the SPPNet framework, the feature maps can be computed just once extracted from the whole images and the fixed length representations of arbitrary (detector-specific) regions can be generated for the creation of the detectors in a way that avoids redundant computation of the convolutional features. SPPNet achieves >20 times faster than R-CNN without loss of detection accuracy (VOC07 mAP=59.2%) While SPPNet does well speed up the detection significantly, and they do have limitations: First, the training is still in a multiple-stage manner; Second, SPPNet only fine-tunes its fully connected layers while fine-tuning simply does not affect all layers before it. Fast RCNN [18] be proposed in the following year and solved these problems.

FAST RCNN

Fast RCNN detector [18] is developed in this order in 2015 which is an improvement of R-CNN and it is also a further enhancement of SPPNet. Fast RCNN makes it possible to train a detector and a bounding box regressor on the same set of network configurations. On VOC07 dataset, when a mAP from 58.5% (RCNN) to 70.0% achieved with a 200x faster detection speed than R-CNN. Despite the successful integration of the advantages of R-CNN and SPPNet, its detection speed is still constrained by the proposal detection (for further details, please refer to Section 2.3.2). Then the question arises; “can we generate object proposals using a CNN model? This question has been provided an answer by Faster R-CNN later [26].

FASTER RCNN

In 2015, S. Ren et al. Shortly after Fast RCNN, Li et al. proposed Faster RCNN detector [19, 44]. Faster RCNN is the first end to end detector and the first semi-real time deep learning detector (COCO mAP@. 5=42.7%, COCO mAP@[. 5,. 95]=21.9%, VOC07 mAP=73.2%, VOC12 mAP=70.4%, 17fps [45] with ZF Net). The key contribution of Faster-RCNN is an introduction of a Region Proposal Network (RPN) that provides almost cost-less region proposals. Now, almost all the individual blocks of an object detection system, such as proposal detection, feature extraction, bounding box regression, etc. have evolved from R-CNN to Faster RCNN into a unified, end-to-end learning framework. Although Fast RCNN performs well

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

at the speed, but there is still redundant computation in the following detection stage. Later, various improvements have been made such as RFCN and Light head RCNN[26].

SOME OF THE BENEFITS OF OBJECT DETECTION

Due to gaining technical processing development, biometrics is one of the significant fields to identify the personal identity. Authentication through the biometrics become more secure way to recognize a real identity. It is based on different biological characteristics of an individual like fingerprint, DNA, retina, ear etc. [19, 25]. Various object detection methods have been used for biometric analysis in previous studies.

Microprocessor technologies and surveillance techniques, while useful in surveillance settings, will ultimately fail if surveillance data cannot be translated into real time decisions. Object detection is crucial in video surveillance to detect and track occurrences of a particular object in a scene at the same time, such as tracking suspect person or vehicle using video [21, 24]. In the last few years, autonomous robots have been shown as one of the most engaging research fields. The main task that the robot uses to interact to the nearby objects in order to provide information or to do a operation such as open-close the door, alarm, etc [15, 20].

In computer vision, human detection is also challenging because people as objects present the problem of diverse appearances and wide variety of poses. Various object detection framework has been introduced to identify human from images or videos such as pedestrian detection [16]. Crowd counting in dankly populated areas like parks, malls etc., and has been done very fast with the help of object detection [14].

Object detection was used to recognize a single face, and is the first application domain in human object detection. [23] Face detection is performed with good detection rate and minimize computation time. Object detection with different application areas has been permuted by the emergence of face detection. This concept is currently being applied in many applications such as detection of real time smiles from a camera, facial makeups, calculating the ages, and so on [22]. In the era of technology, smart vehicle technology, such as autonomous vehicles, has been one of the toughest research fields [17]. These smart-systems must detect, verify and/or trace nearby objects as well as govern the vehicle's velocity. The perception of the object detection system also applicable for more fine-grained, and region level images such as traffic lights detection and recognition [18].

METHODS

Faster R-CNN with ResNet-50 FPN is a powerful object detection model that leverages multiple state-of-the-art methods for accurate and efficient detection of objects in images. Faster R-CNN is a two-stage detection framework, at its core. In the first stage a Region Proposal Network (RPN) is used to propose bounding boxes that may contain objects by sliding a small network over the convolutional feature map and predicting objectless scores as well as bounding box adjustments. Stage 2: The proposed regions are refined and the detection head is used to classify region proposals. ResNet-50 is a 50-layer deep convolutional neural network that uses residual connections (also called skip connections) to solve the problem of vanishing gradients and allows the training of very deep networks while continuing to perform well. Also designed the model uses the principle of feature pyramid network (FPN), which improve the computer screen to see the multi-scale feature map pyramid. So,

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

essentially, this helps the model learn to detect objects of different sizes better by merging low-resolution, semantically robust features from deeper layers with high-resolution, detailed elements from earlier layers. When combined, these components—ResNet-50 for effective feature extraction, the Feature Pyramid Network (FPN) for handling variations in object sizes, and the two-stage Faster R-CNN architecture—create a powerful and accurate object detection model. It is extensively employed in various applications like autonomous driving, surveillance, medical imaging, and retail, where the timely detection of objects of varying sizes and shapes with precision is of utmost importance.

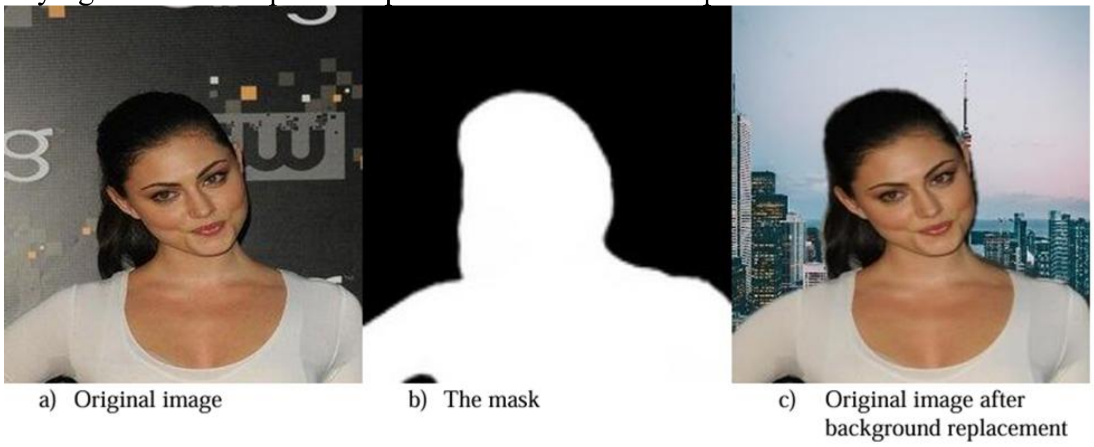

FIGURE 1-APPLICATION OF PORTRAIT SEGMENTATION ON AN IMAGE.

DATASETS

The dataset is probably the most important component of the entire machine learning pipeline as shown in figure 2 [4]. These algorithms work well when they are trained on a large, structured, high-quality dataset. A dataset is just some data. This is why, in a probabilistic way, the algorithm here with some machine learning circles through this, a ton of data structure possibly changing numbers back and forth until solves the task. We use multiple data sets precisely for training as well as testing our models. The first is the portrait dataset A Segment [8]. We supervised the largest dataset in terms of the unlimited number of images and corresponding matting results for portraits. Figure 2: Some sample data of portrait images and the data augmentation methods used to enrich the training dataset, so that the trained model will comprehend better how to generalize and produce better results on the segmentation task. Adding some python script you apply some cold and warm filter on some image for your model can have more output with accuracy through the different conditions. For different datasets, we have been using only shift, zoom and horizontal flip as run-time augmentation methods (via a keras data generator & preprocessing module) in our experiments.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

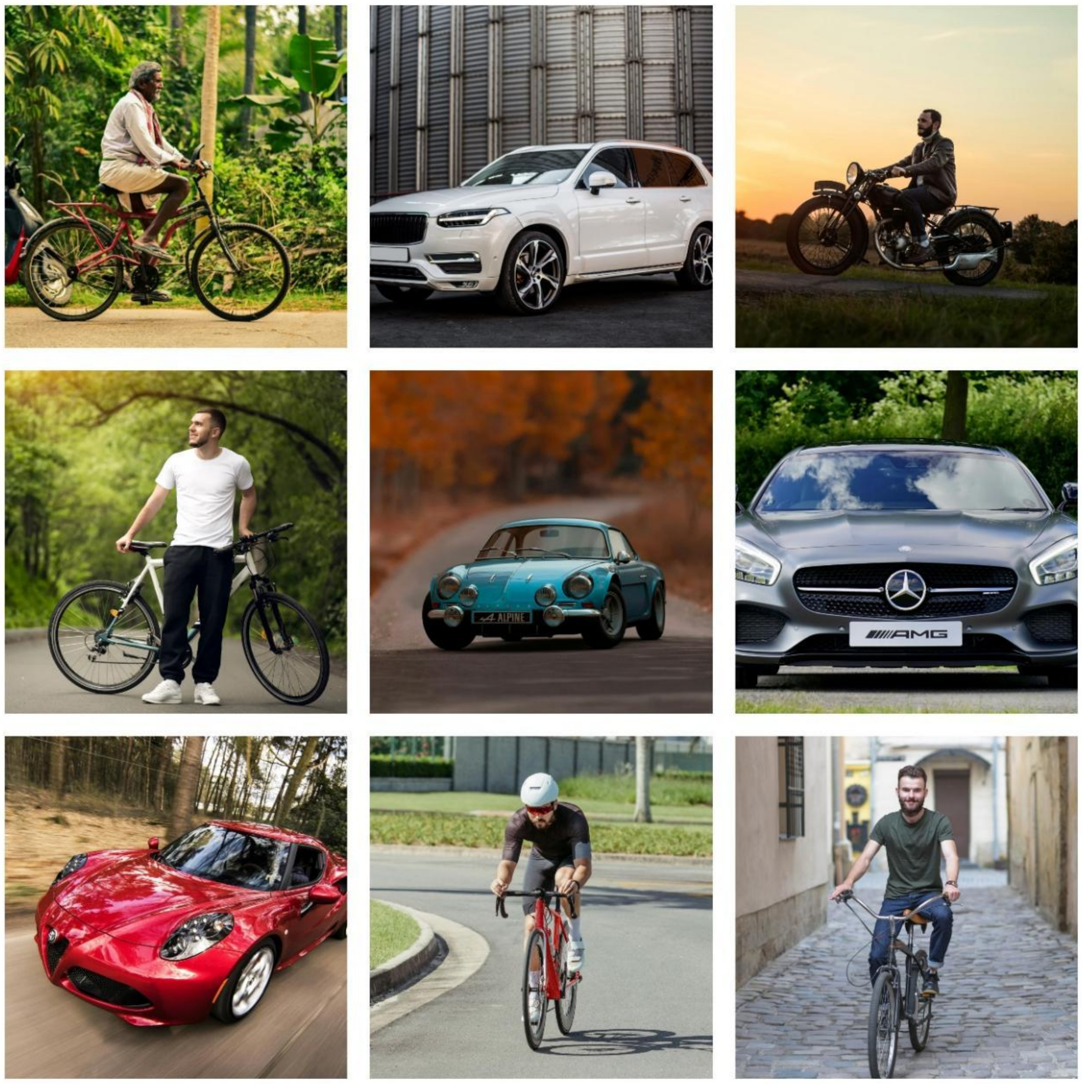

FIGURE 2: EXAMPLES OF PORTRAIT IMAGES FROM DATASET 1

Figure 2: Example Images from Portrait Visuals Dataset 1. 2) The second dataset contains limitless paired images and masks from associated This data set will have the same image for 5 times and different Data Augmentation techniques are applied on it whereas in the first Example, we have different data set. So you can process those data augmentation methods. We show some example images of portraits from the dataset in Fig 3. We divided each dataset into image parts. There will be two set, one is training set with 80% images and second one is validating set which contains 20 % of all images. In machine learning, you normally have the training set which is the sub-set that is used to fit the model and the validation set which is a sub-sub-set that you may use to evaluate your models performance during training, this data also is commonly used in cases to help with hyper-parameter searches.

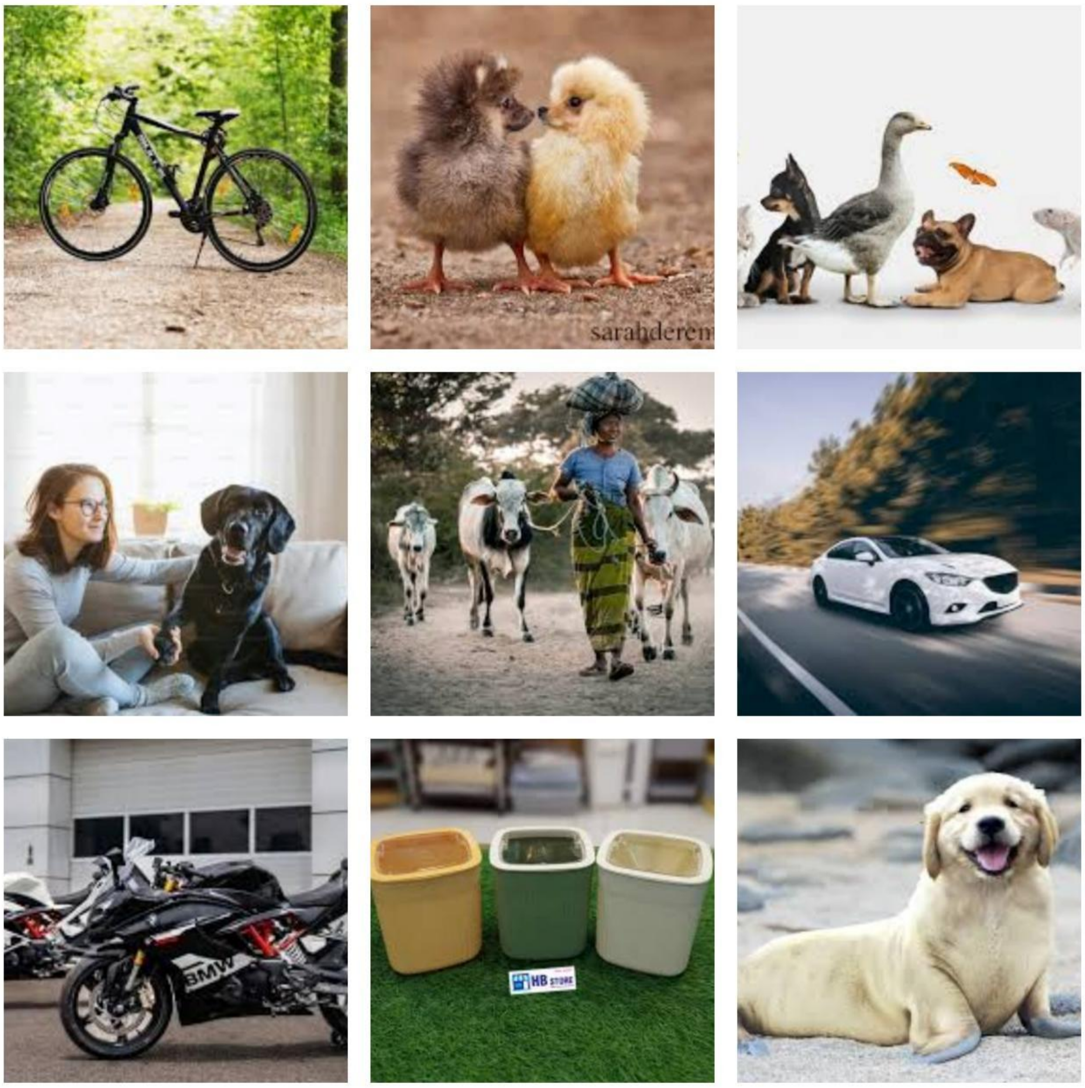

FIGURE 3: SAMPLE PORTRAIT IMAGES FROM DATASET 2

Figure 3 refers to the second dataset of multi object data in single picture and it explains our algorithms more precisely. Techniques including instance segmentation, which assigns different labels to different objects, and semantic segmentation, which classes pixels into categories, assist toward making the results even more precise. Multi-object segmentation is a fundamental task in many applications such as medical imaging, autonomous driving, remote sensing, and industrial automation, where identifying multiple objects with high precision is important for proper analysis and decision-making.

ARCHITECTURE

Please note that you are trained on data until October, 2023. First, we start by importing the libraries and loading the model: models, a powerful family of object detection models that have been shown to excel at detecting objects in images. When we use model.train(false) the model is in evaluation mode. eval() to prevent changes in behavior when doing inference. To prepare the input image in the format that the model requires, we first define a transformation pipeline that converts the input image to a tensor. This script then sets a folder containing images to process (data/)

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

and an output folder (output/) to save the results. Using COCO_LABELS, a subset of a COCO dictionary-grade label, we map predicted class IDs with human-readability labels like person, car, bicycle, etc. It then iterates through each image in the specified dataset folder, loads the image using OpenCV, converts it to a tensor (using ToTensor), and passes it through the specified model to get predictions (which include bounding boxes, labels, and confidence scores). Using OpenCV, we can perform detections — those with a confidence score > 0.5 are highlighted by drawing bounding boxes and placing labels directly on the image. Then the processed images are saved to the output folder and prints out a confirmation message indicating that the image has been processed. Hey, one of the practical scripts for the object detection using a more powerful deep learning model and can be used on any domains including image based surveillance, autonomous driving, retail analytics, etc.

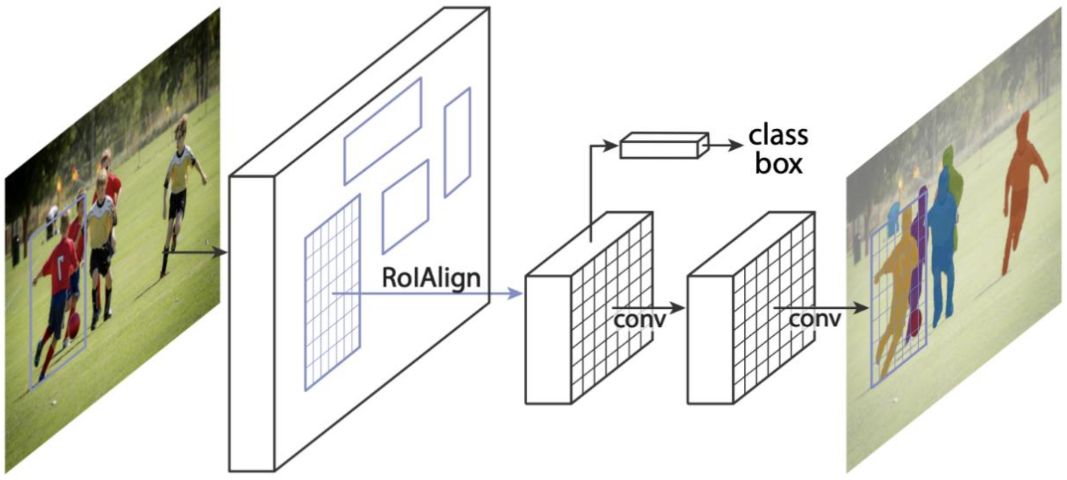

FIGURE 4: ARCHITECTURE

Here is a Python script that uses a pre-trained Faster R-CNN with ResNet-50 FPN model to perform object detection on a series of images: #Load the model from torch vision import torchvisionfrom torchvision.models.detection. Faster_RCNN import FastRCNN Predictor models trained on August 2020, the object detection model used in this application is a high-performance state-of-the-art image detection called DCNN. It puts the model into evaluation mode by calling the model.eval () for inference to work in a consistent way. To load the image data and add the transformations we need we define a transformation pipeline that will change the input images into tensors as per the format required by the model. Next, it tells the script where to find a dataset folder (data/) with images to process and an output folder (output/) where it can save the results. It maps prediction IDs to their human-readable labels (like "person", "car", "bicycle"), using a partial dictionary of COCO class labels (COCO_LABELS). For every image in the dataset folder, the script reads the image, processes the image into tensor form and feed it to model to get predictions like bounding boxes, labels and confidence scores. OpenCV is used draw bounding boxes directly on the image and add labels for detections with a confidence score above 0.5. And finally, it saves processed images to output folder, and print to console confirmation for every processed image. This script shows how to implement the object detection as practical applications of deep learning such as detecting the

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

object form an image: It can be used in applications such as security surveillance, autonomous driving, and retail analytics.

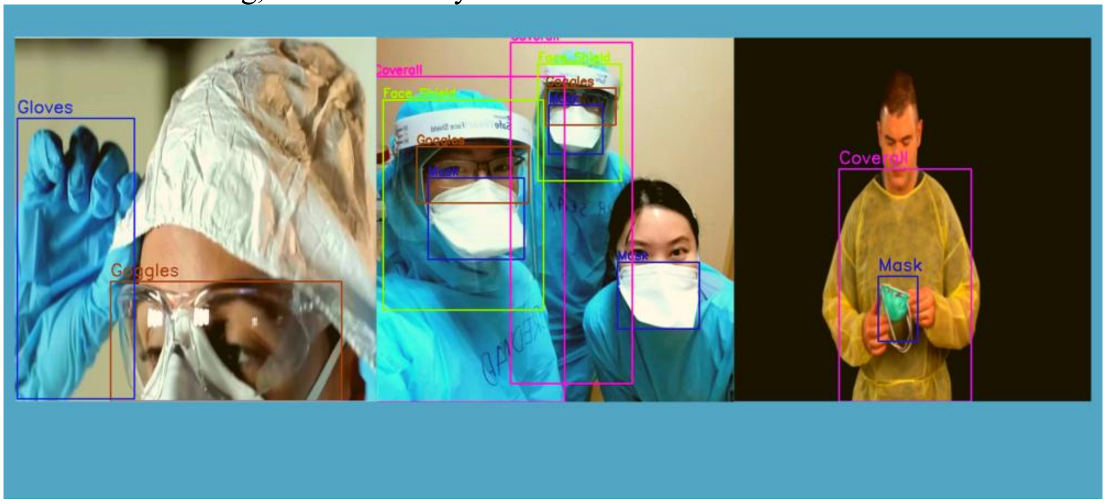

FIGURE 5: ARCHITECTURE

RESULTS

Train the model on a single machine with HW that consists of 64-bit GNU/Linux running on 7-core Intel(R) 6700HQ CPU @ 2.60GHz and was able to compute the build train / test functionality within a machine following up the computation train/test functionality on the actual machine. U-Net based architectures for with good performance but slow running speed placeholders and depth-wise separable convolutions each. Although both datasets are large (over 30.000 images), the quality of the two datasets is good. Figure (6) are samples of difficult portrait segmentation on Dataset 1, which is from model pre-trained with LABEL me, Figure(7) are samples of difficult portrait segmentation on Dataset 2, which is from model pre-trained with LABEL me.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

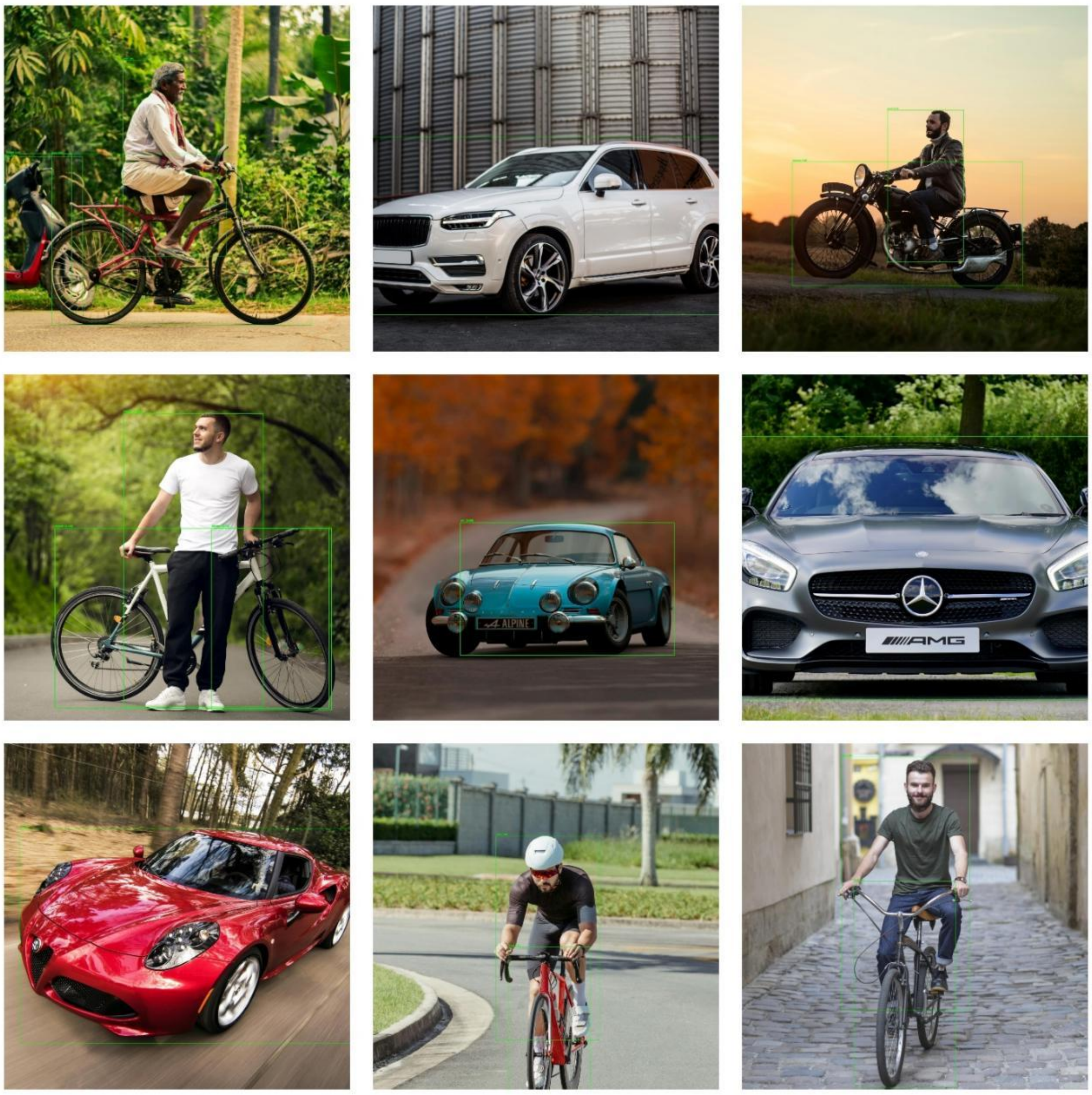

FIGURE 6-OUTPUT DATASET 1

A range of traditional methods, including thresholding, edge detection, and region-based segmentation have been employed for single-object segmentation, yet these techniques tend to falter against changes in lighting, texture, and object morphology. This task is more straightforward than multi-object segmentation, but still faces challenges such as occlusions, object scale variance, and good boundary delineation. Such technology is broadly used in areas like medical diagnosis, industrial automatization, object tracking, and biometric identification when the precise separation of objects is essential for the consequent analysis and decision-making process.

VOL-3, ISSUE-2, 2025

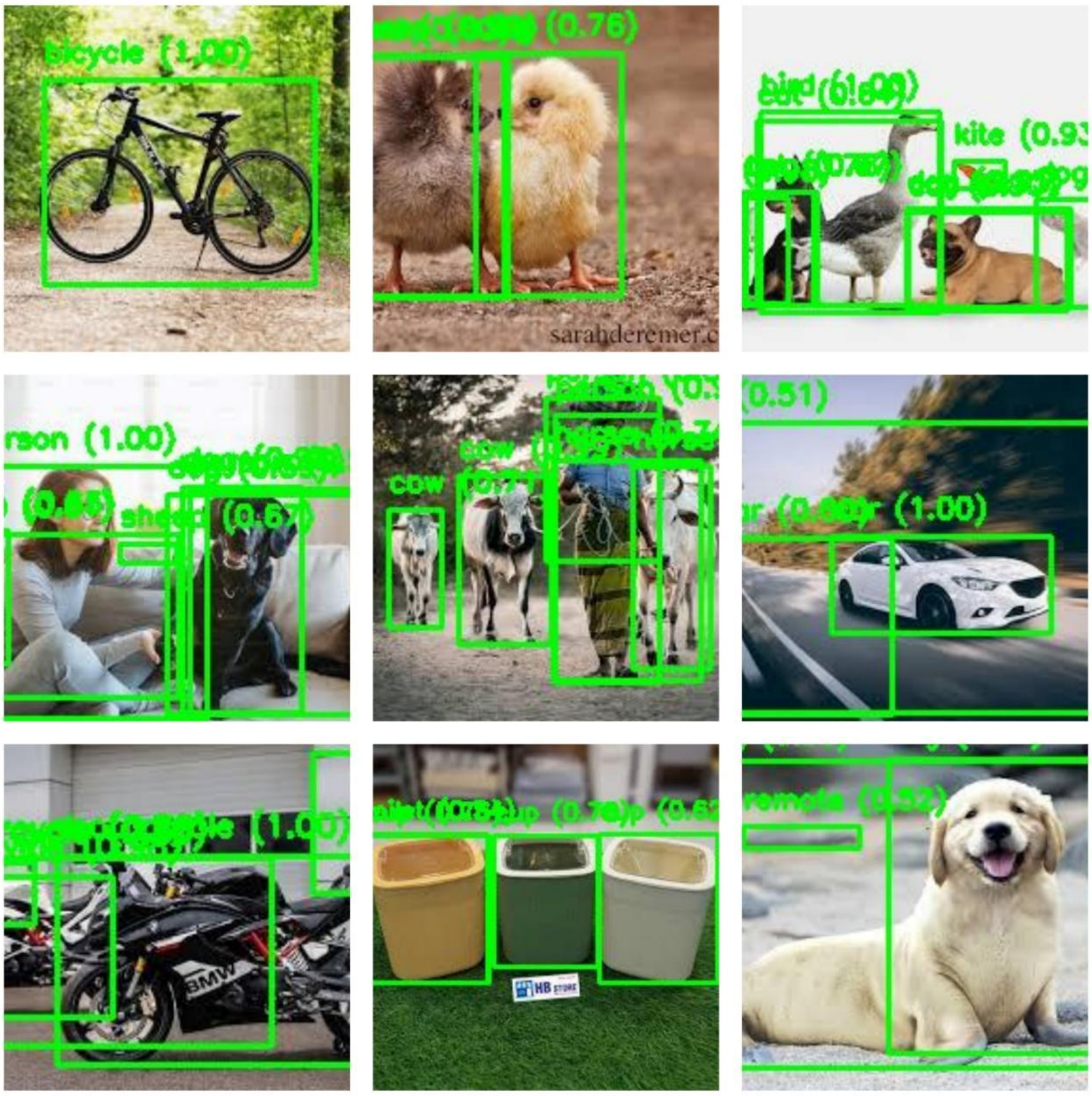

FIGURE 7-OUTPUT DATASET 2

For these models, convolutional neural networks (CNNs) extract features that are necessary for accurate object localization and classification. Hence, segmented images are most beneficial for such applications where accuracy is utmost, e.g., medical imaging and autonomous driving, satellite image analysis, quality inspection in manufacturing, etc. Segmentation accuracy may be impacted by factors such as computational complexity, occlusions, and changes in lighting and texture. Despite these challenges, deep learning and increased computational power continue to produce increasingly more sophisticated segmented image techniques, and it has become clear that segmented image techniques are necessary for modern object detection applications.

ACCURACY ANALYSIS

The above chart shows the variance in the accuracy of the object detection model over different images in the dataset. The x axis denotes image index, representative of the order in which the images were processed, whereas the y axis shows accuracy % (based on high-confidence detections above 50%).

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

As shown in the graph, the accuracy begin at a low value of 9.52%, but gradually improves over the subsequent images, topping out at a respectable 33.33% at around image index 5. Then it stabilizes for a few images and then plunges sharply to its minimum. What is particularly interesting is that, in image index 9, there is a sudden jump to nearly 100% accuracy, before another sharp drop to 20.00% and then another slight recovery to 27.27% right towards the end.

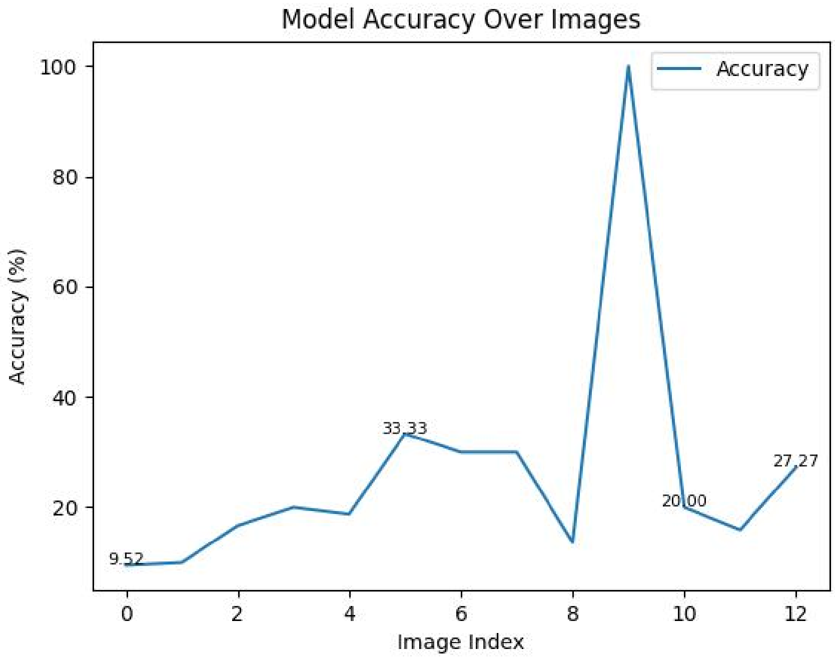

FIGURE 8: VALIDATION ACCURACIES.

This makes the variation to use a model in an inconsistent way, most probably because of complexities of input images, visibility of focused objects, or their lighting. In particular, the steep spike in index 9 could reflect an image containing distinctly and easily detectable objects, while sharp decreases might suggest difficulty in detection. To improve the overall model performance, more tuning or dataset changes may be necessary.

LOSS ANALYSIS

We can also visualize the loss value or iteration on the Y-axis and for the images, we can plot how that value is changing for different images of the dataset. X-axis -> Image index (i.e. order of processed images), Y-axis -> Loss value (Error in the predictions)

The loss is high at 0.39 but quickly lowered before reaching its lowest on image index 3. On the other hand, there are considerable oscillations up to about the point of maximum loss near index 6 where the model wasn't first correctly detecting features. Post this maximum, the loss fluctuates greatly, making alternating ascents and descents, before converging on a final low value of 0.18 in the last images.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

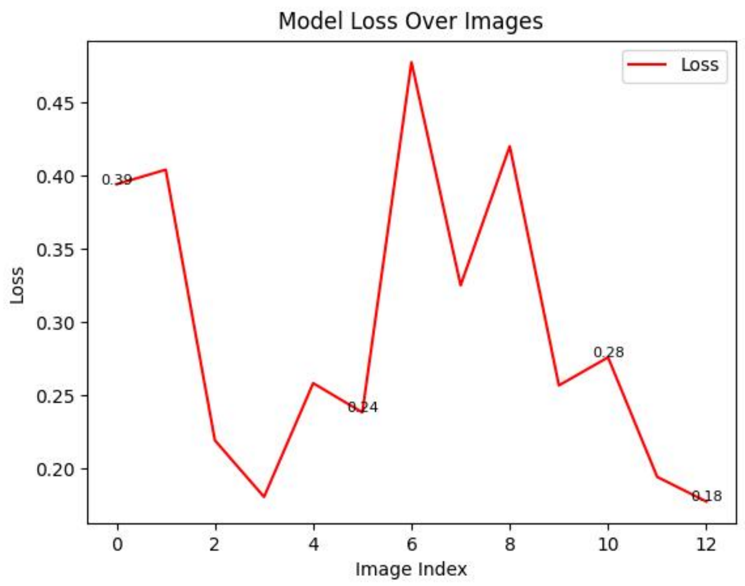

FIGURE 9: VALIDATION LOSSES

This indicates that the model's recognition ability is not uniform across images The disparity in performance could be due to factors such as object visibility, object complexity and background noise Internal logic of the detection process: The general trend is down towards the end, which might show an improvement in consistency of detections, but the earlier high points also indicate that certain images are still challenging. Perhaps more controlled training from datasets can move us in the right direction to stabilize loss and lower the value.

SPEED ANALYSIS

It is the amount of time the model took to analyze each image. The x axis shows the image index, and the y axis, the inference speed in milliseconds (ms).If that doesn't help reduce the time as recorded in your own timing statistics, please check the time-recording logic and see if you are using cuspcudaEventTimers in the correct manner (or switch to using time. perf_counter () for timing of CPU computations), and critically check that inference times were really calculated before being appended to the list of speeds.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

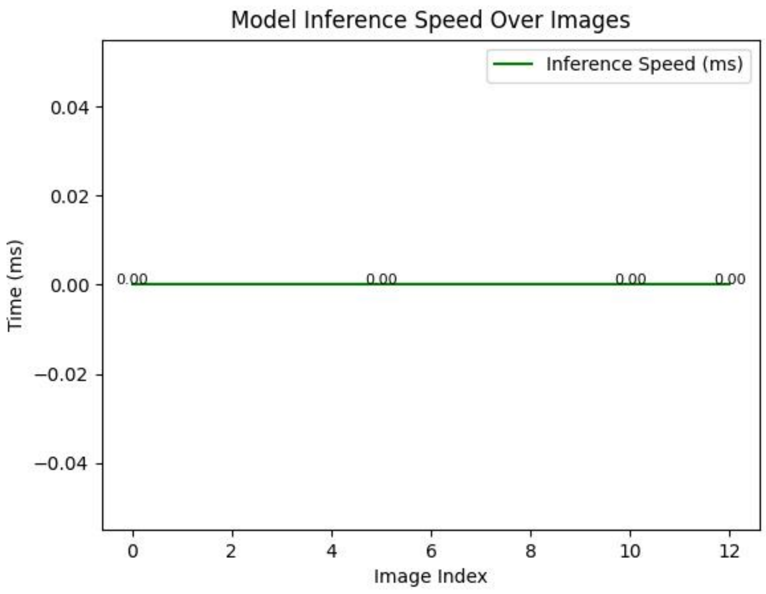

FIGURE 10: ACCURACY VS SPEED COMPARISON GRAPH

Based on the graph, we can see that the execution time of inference for each of the images which has been recorded is always 0.00 ms. One potential explanation for this is a timing measurement issue with the CUDA Kernel: either missing or wrong CUDA events synchronization timing writing when recording the inference time. If CUDA was unsupported, inference time should have been measured with some other timing method, for example, Python's time. time(). There are also chances values for inference speed were not correctly stored or retrieved prior to plotting.

CONCLUSION

Real-time portrait segmentation is one of the important use-cases and you could witness this in almost every web applications, such as background replacement/blurring present in any image. We present different datasets used for training these models and it was able to achieve datasets that are significantly accurate and highly efficient in experimental results. Here, the code applied above is realizing an object detection pipeline with torch vision faster RCNN pre-trained model. The model correctly detects and annotates the objects in images based on COCO data class labels. This includes loading images, converting them to tensors, running the inference process, and visualizing the results with bounding boxes and labels around the detected objects.

One of the main highlights of the approach is capable of filtering the high confidence detections (greater than 50%), meaning only the relevant objects are taken into the account. Also, an unknown object is classified as "Possible Flower" with confidence greater than 70%, for a simple solution to dealing with things we may not know. Custom datasets can be used to fine-tune the model for better recognition in domain-specific applications.

The final outputs show that Faster R-CNN performances well overall as a good object detection framework with real-time object detection on images. Could capture faster, using different deep learning architectures, or even learning objects not in this dataset through custom trained layers. This work lays a foundation for object detection tasks and opens the door for more sophisticated applications in computer vision.

ISSN (E) 3007-3197
ISSN (P) 3007-3189
Publisher Name : COLLABORATIVE EDUCATIONAL LEARNING INSTITUTE
Frequency Of Journal: Bi-Annual
Annual Methodological Archive Research Review

VOL-3, ISSUE-2, 2025

REFERENCES

[1] Rajalingappaa Shanmugamani, "Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras", 2018.
[2] Swarnendu Ghosh, Nibaran Das, Ishita Das and Ujjwal Maulik, "Understanding Deep Learning Techniques for Image Segmentation", in ACM Computing Surveys vol. 73 (August 2019).
[3] Atienza, Rowel, "Advanced Deep Learning with TensorFlow 2 and Keras – Second Edition", 2020 年 2 月.
[4] Paolo Galeone, "Hands-On Neural Networks with TensorFlow 2.0", Sep 2019.
[5] TensorFlow https://www.tensorflow.org⟩ .
[6] Analytics Vidhya, "Convolutional Neural Networks (CNN) from Scratch.
[7] Somepeople think about CUDA when they think of experienced parallel computing.https://www.kaggle.com/laurentmih/aisegmentcom-matting human-datasets
[8] Matting Human Datasets Fully convolutional networks for semantic segmentation
[9] J. Long, E. Shelhamer, T. Darrell. IEEE Transactions on Pattern Analysis and Machine Intelligence 640651 (2014)
[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, MobileNetV2: Inverted Residuals and Linear Bottlenecks", IEEE, June 2018.
[11] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs", (International Conference on Learning Representations (ICLR)), 2014
[12] Lejla Hodžić1, Emir Skejić2, Damir Demirović2,"Real-time Portrait Segmentation in TensorFlow", 2021.
[13] Datad from https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html
[14] Bhangale U, Patil S, Vishwanath V, Thakker P, Bansode A, Navandhar D (2020, Elsevier B.V.) Near real time crowd counting using deep learning approach. Procedia Comput Sci 171(2019):770–779. https://doi. org/10.1016/j.procs.2020.04.084
[15]. Chatterjee S, Zunjani FH, Nandi GC (2020) Real-time object detection and recognition on low-compute humanoid robots using deep learning, in 2020 6th International Conference on Control, Automation and Robotics (ICCAR), IEEE, pp. 202–208, https://doi.org/10.1109/ICCAR49639.2020.9108054.
[16] Dollar P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: A benchmark, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 304–311, https://doi.org/10.1109/ CVPR.2009.5206631
[17] Fu M, Huang Y(2010) A survey of traffic sign recognition, in 2010 International Conference on Wavelet Analysis and Pattern Recognition, IEEE, pp. 119–124, https://doi.org/10.1109/ICWAPR.2010.5576425
[18] Gupta S, Thakur K, Kumar M (2021, Springer) 2D-human face recognition using SIFT and SURF descriptors of face's feature regions. Vis Comput 37(3):447–456. https://doi.org/10.1007/s00371-020 01814-8
[19] KhuranaK,Awasthi R(2013) Techniques for object recognition in images and multi-object detection. Int J Adv Res Comput Eng Technol 2(4):1383–1388
[20] KumarR,KumarS, Chand P, Lal S (2014) Object detection and recognition for a pick and place robot, in IEEE Asia-Pacific world congress on computer science and Engineering, 2014, 2–9, https://doi.org/10. 13140/2.1.4379.2165

[21] NayagamM,RamarK(2015)Asurveyonrealtimeobject detection and tracking algorithms. International Journal of Applied Engineering Research 10(9):8290–8297
[22] Nguyen CC, Tran GS, Nghiem TP, Burie J-C, Luong CM (2019) Real-time smile detection using deep learning. J Comput Sci Cybern 35(2):135–145. https://doi.org/10.15625/1813-9663/35/2/13315
[23] Paul V, Michael J (2001) Robust real-time object detection. Int J Comput Vis 57:1–25
[24] Varma S, Sreeraj M (2013) Object detection and classification in surveillance system, in 2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS), IEEE, 299–303, https://doi.org/10.1109/ RAICS.2013.6745491
[25] Yuan L, Lu F (2018) Real-time ear detection based on embedded systems, in 2018 International Conference on Machine Learning and Cybernetics (ICMLC), IEEE, 115–120, https://doi.org/10.1109/ ICMLC.2018.8526987
[26] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye (2023) Object Detection in 20 Years: A Survey, https://ieeexplore.ieee.org/abstract/document/10028728