Near Real-time Forgery Detection and Localization in Stereo RGB and 3D LiDAR Data from Autonomous Vehicles

Somayeh Mohammadpour, Sagnik Dakshit, Balakrishnan Prabhakaran

***Abstract*—The widespread availability of applications for forging multimedia data bolsters the need for their detection and localization especially in autonomous decision-making systems such as autonomous vehicles. Traditional and machine learning methods have been researched extensively for the detection of the same in 2D images but are scarce for the 3D point cloud. Recent trends have seen a shift to leverage the power of deep learning. While recent studies have proposed deep learning-based architectures for accurate detection of forgery in 2D, studies in detection and localization in 3D, in general, are still in preliminary stages. We propose a Forgery Detection and Localization network (FDL-Net) that accurately detects in 2D and localizes the forgery simultaneously in 2D and 3D point cloud. Such a Vision Based Measurement (VBM) system can compute the probability of forgery for each pixel in the image and predict the tampered area ultimately. We train FDL-Net on groups of Easy, Medium, and Hard automatically-generated attacks on RGB and point clouds, based on the KITTI Object Recognition dataset. FDL-Net is able to detect and localize forgery with a high Intersection over Union scores of 0.9773, 0.9324, and 0.73451 for each group respectively and localize the region of attack in 2D stereo RGB and 3D point cloud in less than 300 milliseconds. In comparison to current state-of-the-art architectures, FDL-Net is superior in its ability to detect and localize not only Easy, Medium but also Hard attacks that are not visible in most cases to the naked eye. To show that our proposed end-to-end network can be a general approach to segment the forged area within the streaming data, we compare its performance with other state-of-the-art methods on benchmark datasets such as CASIA V1.0 and CASIA V2.0.**

***Index Terms*—Autonomous Driving, convolutional neural networks, Forgery Localization, Forgery detection, 3D point cloud, Computer Vision, Self-driving Vehicles, Attacked Dataset**

I. Introduction

As enthusiasm and demand for autonomous vehicles and shipment robots grow, the need for a system to authenticate the data captured by these self-driving vehicles becomes urgent to address. To be able to navigate, such a system has to collect data to sense the environment around them. The sensors, installed on them, are a pair of stereo RGB cameras that collect the 2D information (as left and right views) and 3D LiDAR (Light Imaging Detection and Ranging) sensors that collect the depth information by the concept of time of flight. The latter captured information (in 360 degrees) is considered as an unstructured 3D point cloud where each point is stored as coordinates of X, Y, and Z.

S. Mohammadpour, S. Dakshit and B. Prabhakaran are with the Department of Computer Science, University of Texas at Dallas, Richardson, TX, 75080 USA (e-mail: somayeh.mohammadpour@utdallas.edu; sdakshit@utdallas.edu; bprabhakaran@utdallas.edu).

For safer operations of such a system, typically remote human operators are employed to observe and track the behavior of the autonomous vehicle by streaming the stereo camera and the LiDAR data. During this streaming of data over a network, the data from the autonomous vehicles are potentially vulnerable to attacks. Hackers can manipulate the data on the transmission and mislead the vehicle and the human operator in the loop. To prevent and minimize the risk of the tampered data, an authentication model is necessary to identify the forged data and enable the autonomous vehicle to make the best decision in near real-time, which represents essentially an approximation of time performance of our test cases.

As discussed in [1] [2], vision-based measurement has been leveraged in various automated applications from analyzing the state and intention of the drivers by monitoring their face and predicting any potential human errors to eliminate the accident chance [3] [4] or sensing the environment, detecting objects and finding the path by robots [5] [6] to counting the calories and detecting nutrition of food by analyzing the meal picture [7]. Following the requirements mentioned in [2], our work is a VBM system involving data collected from vision sensors installed on autonomous vehicles, creating an appropriate dataset, and proposing a deep learning approach to authenticate vision data used for measurement and monitoring of autonomous vehicles/robots.

A. Proposed Approach

In this paper, we propose a novel neural network approach, FDL-Net (Forgery Detection and Localization Network), to detect and localize the forgery without any pre-processing or post-processing step. Our proposed Vision Based Measurement (VBM) model not only detects forgery very accurately but also localizes it in 2D and 3D in near real-time. Likewise the human eyes that may visually detect and localize the tampered area by evaluation of the entire image to recognize different shades, sharp borders, or even different quality around the tampered area, FDL-Net learns such above mentioned differences by detecting statistical anomalies introduced at the pixel level. In this way, if the digital forgeries do not leave any clue for tampering and consequently human eyes are not able to catch it, there is still this chance that our approach detects it due to the altered statistics. By extracted discriminative features, the model can measure the probability of forgery for each pixel and localize the forged area at the

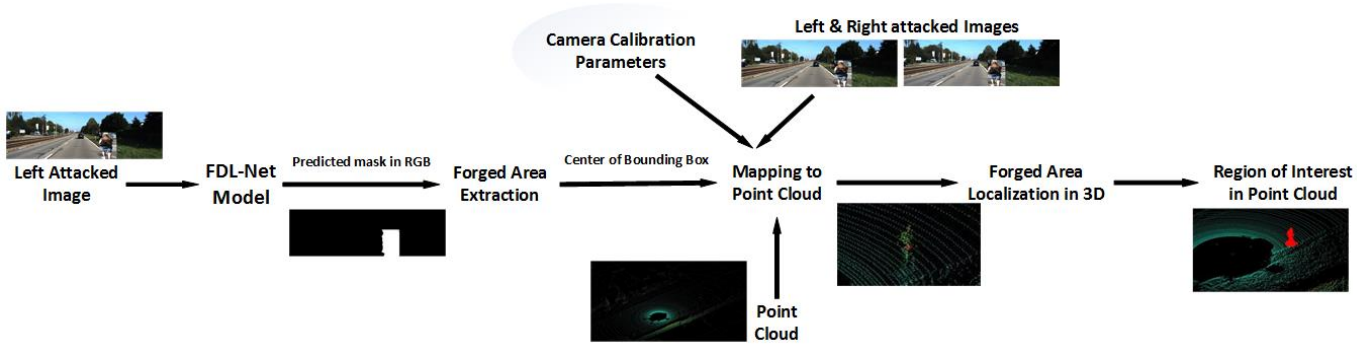

Fig. 1. Our Proposed Flow Diagram.

image level in near real time. The perceived level of danger for an autonomous car depends on the distance. The reaction time to detect and localize the attack is required to be much lower for attacks that are closer to the car making near real-time a crucial requirement. On the other hand, the farther attacks that manifest as smaller regions of attack require an accurate model as they are hard to detect.

The FDL-Net is a U-Net [8] style network to extract more robust features with respect to more complicated input data. The architecture of the vanilla U-Net that is originally designed for biomedical image segmentation, consists of two paths, contracting (or encoder) and expansive (or decoder). The contracting path, used for feature extraction purpose, is built of several convolutional layers followed by downsampling layers (that can be considered as an issue in segmentation task as we discussed later). In the expansive path, used for predicted mask construction, they have an up-sampling step followed by a convolutional layer that leads to halving the number of feature channels and then be concatenated with the corresponding feature map extracted from the contracting path. In the last layer, there is a convolutional layer to classify each pixel based on the final feature vector.

Convolutional Neural Network approaches leverage the down-sampling layers in the architectures to reduce the computational cost however, down-sampling layers lead to loss of valuable information. For instance, in a tampered image that contains a small area under attack, the ratio of forged pixels to non-forged pixels is very small. Therefore, the down-sampling steps in the network can lead to loss of information such as forged pixels resulting in the failure of the model.

In FDL-Net, we chose a different backbone (which is not including any down-sampling step) to extract the robust features of the input data and used a hybrid loss function to diminish the effects of the imbalanced dataset. The model obtained by FDL-Net is used to localize the area under attack in stereo RGB images. Then, we are able to map the extracted area in 2D to 3D point cloud data by camera calibration parameters and 3D K-Nearest neighbors.

Figure 1 shows the operation sequence of the proposed approach to localize forged area in 2D RGB images and 3D point cloud data. Having detected and localized forgery in RGB images, we proceed to localize the same in the 3D point cloud. The segmentation mask obtained from the FDL-Net model is essentially a localization of the forged area in 2D. The center of the segmentation mask is computed from the extracted bounding box of the segmented region. The center point presenting the forged region in 2D is then mapped to the point of interest in 3D. We then compute the region of attack in the 3D point cloud by running a 3D K-NN search. The details are furnished in later sections.

To train the FDL-Net, we utilize the framework introduced in ADD-FAR [9] to create a dataset based on the KITTI 3D Object Benchmark Suite [10] automatically. This attacked dataset meets our requirements and consists of tampered 2D RGB images and ground truth segmented masks, corresponding 3D point cloud data and camera calibration parameters which used for forgery localization in 3D point cloud data.

Our Contributions The proposed approach works in near real-time: the average time for forgery localization is around 66 milliseconds giving the ability to raise alerts about the forgery almost immediately. The localization of the detected forgery in 3D LiDAR data, run on a machine without GPU, takes around 200 milliseconds (due to the voluminous nature of the point cloud data) and can facilitate near real-time process for subsequent threat evaluation and mitigation. Experimental results show that the proposed FDL-Net can identify forged pixels in RGB images even when the ratio of the forged pixels to the entire scene's pixels (i.e., *forgery ratio*) is very low. FDL-Net can detect and localize forgery with high IoU (Intersection over the Union) scores of 0.9773, 0.9324, and 0.73451 in images with forgery ratios of 0.06 (Easy attacks), 0.026 (Medium attacks), and 0.005 (Hard attacks).

II. RELATED WORK

Manipulation of multimedia data is primarily of two types namely Steganography and Tampering and their detection methods can be categorized as proposed by A. Piva et. al [11] into Active and Passive Approaches. The Active Approaches have some source information and use methods such as Watermarking and Digital Signatures. For instance, Bahirat et. al [12] proposed a watermarking based framework for Authentication and Localization of tampering in RGB and 3D point cloud. On the other hand, Passive Approaches also referred to as Blind Forensics [13] primarily deal tampered data where source information is not available. While all the methods have been progressively getting better, deep learning

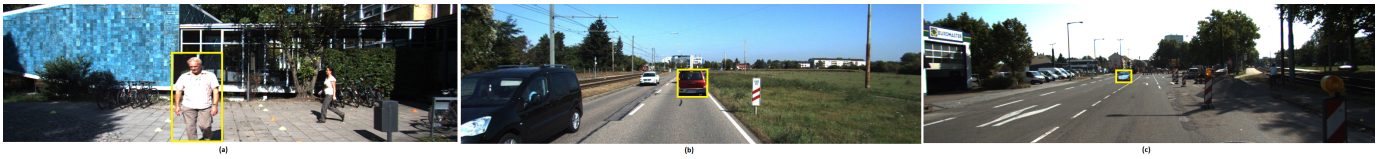

Fig. 2. Examples of each type of attacks in our automatically generated dataset (forged KITTI dataset) a) Easy, the man approaching to the camera is the forged object in the scene, b) Medium, the red SUV driving ahead of the autonomous vehicle, and c) Hard, forged object is the silver car, far from the autonomous vehicle. Note that the tampered region is indicated by bounding box in yellow and the number of forged pixels is 32332, 6080, and 1344 out of over 450K for each type, respectively.

based tampering detection approach handles the drawback of having to try various forensic tests to understand how the image has been tampered and also the needs to balance false positives and negatives of those tests as pointed out by B.Bayer et. al [14].

This drawback motivates the need for tampering detection and localization techniques for automated systems. Many deep learning architectures have been proposed aimed at building better models, some require extensive feature engineering [15] while others introduced new layers for better feature learning [14], new architectures such as triple network with conditional random fields [16], and auto-encoder based feature extraction and labelling [17]. For different deep learning architectures, various training methods have also been proposed such as transfer learning [18], extracting features through deep networks followed by traditional machine learning for classification [19], patch based learning especially when the dataset is small [14][15][19][20]. The patch based approach though successful in achieving high accuracy involves a lot of pre-processing and is prone to errors during relabelling the patches. The overhead of pre-processing keeps the system from being near real time. The above discussed methods though successful in detecting forgery cannot localize the forged area.

B. Liu et. al [21] worked at detection and localization of forgery but the model overfits the data, X. Wang et. al [22] used Mask R-CNN and Sobel edge detection filter to focus on manipulated boundaries and [23] used multiple convolution branches and merge them which has an higher overhead. R. Yancey et. al [24] used two-stream faster RCNN network, one stream takes RGB data as input while the other one takes a noise filter, ELA (Error Level Analysis) that helps to find the area of interest by identifying different compression levels within an image. But, this approach works for JPEG images only. The authors in [25] used another noise stream instead of ELA. It extracts the noise features by passing the image through a Steganalysis Rich Model (SRM) filter that finds the noise inconsistency between pristine and doctored areas. This approach cannot be considered as a general method when the un-tampered and tampered regions captured by the same camera brand and model since they have the same noise specification. The last two recently discussed approaches output the forged area as a rectangular bounding box, not showing the fine coarse border around the region of interest. Without having above mentioned restrictions, our proposed approach has the advantage of near real time prediction and localization on CASIA V1.0 with a higher accuracy than that of Salloum et. al [23].

III. Attack Model And Dataset

With the increasing use of autonomous vehicles, there has been a push towards the need for monitoring them both for security as well as the correctness of decision making. In fact, the State of California has a legislation that requires human operators to remotely monitor the movement of such vehicles during testing [26]. This implies streaming of the video data – typically, stereo RGB camera data as well as 3D LiDAR (Light Detection and Ranging) data – from the vehicles to the remote human operator. This introduces the potential risk of the video stream from the car to the remote operator being hacked. A maleficent system or user can try to manipulate the stream of stereo RGB video frames and/or 3D LiDAR scans and create attacks to misguide the autonomous vehicle. This motivates the requirement to detect and localize forgery in autonomous systems in near real time. The attack can take place on all modalities of data collected by the system during the process of collection, transmission from the car to remote operator. In this paper, we consider an attack model where the stereo RGB images are attacked by introducing new objects into the scene. Then, we utilize a post-processing technique such as blurring to smooth the tampered region contours to make the sharp edges of the attacked area invisible. Based on this attack model, we introduced attacks in the stereo RGB images in the KITTI dataset, as explained in Section III-A. We can classify the manipulated images in the data set into three categories in terms of the distance of the forged object from the autonomous vehicle: (a) Easy, easily detectable forged image that contains forged area closer to the vehicle (It means it contains larger number of forged pixels). (b) Medium type contains those attacked images with forged area, not close nor far from the autonomous vehicle. And (c) Hard category deals with images where the attack is far from the vehicle making it hard to detect (the forged object is farther and smaller due to which the number of forged pixels is much less).

Figure 2 depicts examples of such created RGB images from the dataset. It may seems the area under attack is visually detectable by human eyes in most of the cases in existing datasets, however automated detection of forged areas in such manipulated images and localization of the attacked region in the corresponding LiDAR data, are still a challenge as we show through our experiments. Unlike semantic segmentation that localizes each meaningful object in the scene it needs to focus on the difference between forged and non-forged pixels' distributions to extract the discriminative features.

A. Dataset Generation

The existing tampered datasets, to our best knowledge, are not suitable for our purpose due to the following reasons:

- The contents of the image should be appropriate for outdoor environment such as cars, trucks, pedestrians, roads and, so on. As Khanafer et. al. described in [1] if the dataset used for training a model is not adequately representative of the real world situation, it may leads to systematic effects which increase the uncertainty.
- The dataset should consist of tampered stereo right and left color images, corresponding 3D point cloud data, camera calibration parameters, and ground truth segmentation mask that shows the attacked area in tampered RGB image.

For instance, CASIA V2 [27] as a tampered dataset, consists of 5123 manipulated images, cannot meet our requirements. Thus, this issue has encouraged us to exploit the framework introduced in ADD-FAR [9] to generate automatically the doctored data based on KITTI 3D object recognition dataset [10]. Each instance in this public dataset includes a pair of stereo images (left and right), 3D point cloud data, camera calibration parameters/matrices and training labels of objects in the scene. Moreover, we blur the edges of forged area to conceal or smooth the sharp edges of added object into the scene.

The automatically generated dataset contains different scenarios of attack and each scenario includes a tampered RGB image, its corresponding ground truth mask that shows the forged area in the RGB image, and tampered point cloud data. This attacked dataset is categorized in different levels of risk mentioned in [12] such as Easy, Medium and Hard as defined in Section III.

Note that the same approach is applied for generating both, left and right sides of attacked data on left and right sides of RGB images captured by stereo camera installed on autonomous vehicle.

Totally, our automatically generated dataset (based on left stereo images) consists of 5825 different scenarios, including 1634 tampered data in Easy category, 2200 in Hard category, and 1991 in Medium category. Also, the 2D images are varied in size with an approximated value of 1242×375 pixels. Although human eyes can visually detect the region under attack in the most cases, it's challenging for an automated system to detect, localize and map it to 3D point cloud. Also, it's worth to note that in the process of tampering the data, the attackers cannot create very sophisticated and elaborated forgery in real-time so, our generated dataset is a suitable fit for this application although the FDL-Net can compete with state-of-the-art networks on benchmark datasets. The dataset will be made public after the paper review is completed.

IV. Forgery Localization Challenges

There are several challenges in localizing forged area in 2D images and 3D point cloud data using deep-learning networks:

- We need to detect and localize the attacked area in 2D and 3D point cloud data in near real time.
- The ratio of the forged pixels (forgery ratio) to the entire scene's pixels is small approximately 0.06 for *Easy* attacks. This leads to an imbalanced pixel ratio between the forged and pristine pixels. For a segmentation task, such an imbalance affects the model's performance. This problem manifests on a larger scale in Medium and Hard cases where the ratio drops to approximately, 0.026 and 0.005 respectively.
- Mapping the localized region from RGB to 3D LiDAR data in the form of bounding box poses a challenge owing to the sparse nature of point cloud. The bounding box loses its semantic structure when mapped directly from RGB to Point Cloud.

A. Deep network Architecture choices and challenges

We can consider two types of deep-learning architectures for forgery detection: (a) Classification-oriented; (b) Segmentation-oriented.

a) Classification-oriented Architectures: In order to classify whether an RGB image is manipulated or not, there are a variety of deep-learning architectures for classification, that can be used along with transfer learning. Such networks predict whether there exists any forgery in the image as a whole. However, they are unable to localize the region of image that has been attacked. As an experiment, we selected VGG16 as an image-wise classifier, and trained it by transfer learning technique (Using pre-trained model on ImageNet [28]) with our generated dataset described in Section III-A to predict whether an image is forged or not. Each instance of the dataset is an RGB image, labeled as forged (positive class) or pristine (negative class). The results showed that over 95 percent of the images of test set classified correctly by the trained model. Although it can be considered a valid model to detect the majority of forged images, it could not localize the forged area in the images that were classified as attacked.

b) Segmentation-oriented Architectures: perform image segmentation by classifying pixels into different segments. For instance, U-Net [8] is one such architecture originally designed for biomedical image segmentation. We trained U-Net on our dataset to localize the forged area in the RGB image data. Here, forgery detection is posed as a binary classification task where we need to label each pixel of each image in training dataset as forged or non-forged via a segmented mask. This binary mask shows pixels in two colors. In the training model we used, pixels in white represent forged pixels and those in black represent the non-forged pixels. We trained the U-Net model with the use of Binary Cross Entropy as loss function to do pixel-wise classification. The experimental results show the validation accuracy to be almost 92 percent, i.e., the model can classify 92 percent of **total pixels of all images in test set** correctly.

However, when we generated the predicted mask for the forged area in each single image in the test set, we observed that it cannot find the location of attacks in most of the *Hard* and *Medium* attacked images, though its performance is acceptable in predicting segmentation mask for large forged area in *Easy* type. In other words, the majority of pixels that

classified correctly lay in those attacked images with larger forged area. These experiments show that U-Net can help us segment the forgery area but it needs some improvements to work for all types of attack with a reasonable accuracy. These improvements need to address the following challenges:

- *High resolution images and more complicated background:* The dataset of forged images generated using KITTI dataset has higher resolution and more complex background than biomedical images. Hence, the structure of traditional U-Net introduced in [8] cannot extract robust features, and then generate segmentation mask for our problem. Therefore, we need a proper architecture to learn discriminative features and predict forgery accurately.
- *Small ratio of forged pixels to non-forged pixels:* This resulted in the traditional U-Net model's poor performance on Hard and Medium attacked data in terms of its inability to generate the segmentation mask for the forged area.
- *Imbalanced number of non-forged and forged pixels:* affects the detection and generation of the mask even in Easy attacks.

V. FDL-Net: Forgery Detection AND Localization Network

To identify and localize the forged area in 2D images, we take advantage of the U-Net by modifying the architecture using a backbone structure borrowed from the family of EfficientNets for the purpose of extracting more fine-grained patterns. This backbone is utilized in contracting path of our U-Net style network. Inspired by [29], we select the hybrid loss with contribution of both losses, focal and dice loss to train the FDL-Net. We address the challenges described in Section IV-A and justify the solutions as follows:

1) To handle the challenge of high resolution images and complicated background, we need to choose an architecture for contracting path to be more sophisticated than the one used in standard U-Net so that to be able to extract the more robust features. Intuitively, making the network architecture deeper makes sense because we need to add more layers to increase the receptive field for high resolution images. Moreover, adding more channels helps to capture more sophisticated and subtle patterns on the more complex image. Also, considering the need for a near real time model for localization task, we choose the backbone from EfficientNets [30] family. We selected EfficientNetB4 as a backbone for FDL-Net through experiments explained in Section VI. Note that the expansive path is almost symmetric to the contracting path in U-Net style architecture.
   In EfficientNet [30], the authors apply a search algorithm called NAS (Neural Architecture Search) [31] to find a baseline architecture with less parameters but with higher accuracy than some architectures with more parameters such as ResNet-50 or Inception-v2. Then, by compound model scaling with aspects of depth, width and image size they scale the baseline to larger networks so that the most scaled network, EfficientNetB7 achieves state-of-the-art performance on ImageNet in terms of accuracy but having considerably less parameters. This leads to obtaining a model containing less trained weights and satisfies the aspect of lower inference response time in prediction.
   The building block used in EfficientNets is MBConv, mobile inverted bottleneck, [31] [32], to which they use squeeze-and-excitation optimization [33] that helps to improve performance of state-of-the-art neural network architectures and minimize the computational cost.
2) By training the traditional U-Net with BCE (Binary Cross Entropy) as a loss function, the obtained model is not able to alleviate the poor performance caused by the small ratio of forged pixels to non-forged pixels. This leads to the poor performance on the most of the Hard and Medium attacks. We incorporate *focal loss* [34] that reduces the weight of the contribution of easy examples so that the network focuses more on hard examples. This loss makes the model learn classifying of misclassified pixels correctly and helps to conquer the problem of having very small forged area located in Hard and Medium attacks.
3) As we discussed earlier, the dataset is imbalanced since the forged area is very small in comparison with the entire size of 2D image even in Easy type attacks. To deal with the problem of this imbalance between the number of forged and non-forged pixels in each image, dice loss has a crucial impact by learning the class distribution [35].
   The total loss is formulated as the summation of binary Focal loss and Dice loss:

$$L_{total} = L_{Dice} + \lambda L_{Focal} \tag{1}$$

   where λ is a parameter for trade-off between dice loss and focal loss. In our case, we set it to 0.5 since the model performance improves when we empirically try different values such as 0.3, 0.5, and 1. According to calculation of each of them in [29] dice loss and focal loss (that in our case, it's binary focal loss) are formulated as:

$$L_{Dice} = C - \sum_{c=0}^{C-1} \frac{TP(c)}{TP(c) + \alpha FN(c) + \beta FP(c)} \tag{2}$$

$$L_{Focal} = -\frac{1}{N} \sum_{c=0}^{C-1} \sum_{n=1}^{N} g_n(c)(1 - p_n(c))^2 \log(p_n(c)) \tag{3}$$

   Where,
   True Positives for class c, represented as $TP(c)$,
   False Positives for class c, represented as $FP(c)$, and
   False Negatives for class c, represented as $FN(c)$ are calculated by $p_n(c)$, prediction probabilities of pixel n given class c, and ground truth probabilities for pixel n given class c.
   $TP(c)$, $FP(c)$, and $FN(c)$ (for each image) can be computed as:

$$TP(c) = \sum_{n=1}^{N} p_n(c) g_n(c)$$

$$FN(c) = \sum_{n=1}^{N} (1 - p_n(c)) g_n(c)$$

$$FP(c) = \sum_{n=1}^{N} p_n(c)(1 - g_n(c))$$

α and β used in dice loss are the trade-offs of penalties for $FN(c)$ and $FP(c)$ that set to 0.5 since we intend to make them balanced. C is the total number of classes, that in our problem, are forged and non-forged (or background) and N used in formulas is the total number of pixels in each image.

A. Evaluation Metrics

To evaluate our model performance, we use metrics such as Intersection over Union (IoU) score, that is the intersection of ground truth mask and predicted mask over the union of them, and F1-score formulated as:

$$IoU = \frac{mask_{gr} \cap mask_{pred}}{mask_{gr} \cup mask_{pred}} \tag{4}$$

$$F1_{score} = \frac{2 * TP}{2 * TP + FN + FP} \tag{5}$$

Intersection over Union measures the overlap between 2 boundaries. We use that to measure how much our predicted mask overlaps with the ground truth mask. We set Intersection over Union threshold to 0.5. Also, to balance between Precision and Recall we use F1-score to measure this balance. Precision represents a count of how many of the predicted positives are actually positives and Recall is a measure how many of actual positives the model can catch out of all positively labeled. Precision and Recall in terms of True Positives, False Negatives and False Positives are formulated as:

$$Precision = \frac{TP}{TP + FP} = \frac{TruePositive}{TotalPredictedPositive} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TruePositive}{TotalActualPositive} \tag{7}$$

B. Rationale for Backbone Choice

In this section, we summarize different experimental results to choose a desirable backbone for modifying the U-Net style architecture. We choose couple of architectures to extract the discriminative features for contracting path of our U-Net style network such as MobileNetV2, and some from EfficientNets family (to the extent of our resources capabilities).

Among mobile-sized networks, first we try MobileNetV2 [32] as a backbone for modifying U-Net style architecture then select different architectures from EfficientNets family, such as EfficientNetB0 (the baseline of EfficientNet family), EfficientNetB3, and EfficientNetB4. As shown in Table I, the performance of MobileNetV2 and EfficientNetB0 is not adequate for Hard cases. However, by increasing the number of parameters we observe better performance. Among all four choices, EfficientNetB4, as a backbone for our U-Net style architecture, has a better performance to detect and localize forged area in all types of attacks, especially Hard attacks, thus we select it to use in FDL-Net as a backbone.

Figure 3 shows the architecture of FDL-Net that is a U-Net style network with EfficientNetB4 as backbone.

C. 3D Localization Using FDL-Net Segmentation

As shown in Figure 1, the proposed approach detects and localizes the area of forgery from either the pair of stereo images using FDL-Net model. The segmentation mask (white: foreground or attacked area, black: background or non-forged area) obtained from FDL-Net for the forged area in RGB is forwarded to the Forged area Extraction module where the bounding box of the mask is extracted and the center of the bounding box is computed. Mapping of the estimated four bounding box coordinates in RGB to eight bounding box points in point cloud directly is challenging owing to sparsity and collisions. The sparse nature makes it difficult to represent the semantic structure of the entire object or scene proportionally with distance from the camera. It also requires roll, pitch, yaw angle beyond camera matrices for object orientation and pose estimation that is not estimated by object detection algorithms. To reduce computation, keeping the immediate goal and utility of localization in mind, we propose to compute the center of the bounding box extracted from the segmentation mask output of FDL-Net. The center coordinate is mapped to point cloud using the 3D LiDAR data and camera calibration parameters as shown in Figure 1. The center point acts as a representation of the forged area in stereo images and LiDAR from which we estimate a region of forgery. A pair of calibrated stereo cameras are used to compute the disparity, followed by depth estimation. Depth estimation is done by Triangulation following the principles of epipolar constraints and $z = \frac{f.b}{d}$, where z represents depth, f stands for focal length of the camera, b is baseline and d is disparity. The disparity is calculated as $d = X_L - X_R$ where L and R subscripts represent left and right stereo images respectively. The mapped point (X), is obtained by operation of the depth image on camera matrix as shown in Equation 8:

$$X = P_{image\_to\_rect} * R_{rect\_to\_cam} * (R|T)_{cam\_to\_velo} * Y \tag{8}$$

This operation converts the coordinates of points from stereo camera image plane to point cloud sensor plane. The parameters in matrix $R_{rect-to-cam}$ is used to change to camera coordinate system and $P_{image-to-rect}$ to rectify the coordinates from image plane. R and T are the rotational and translation matrices for changing coordinate system from camera to point cloud sensor. Note that in ALERT[12] they operate on the entire image while we are using the same idea only on the points of interest, represented in Equation 8 as Y.

To localize the region of interest in the point cloud, we run a 3D K-Nearest neighbours search. The forged area in the

TABLE I
MODEL PERFORMANCE WITH DIFFERENT BACKBONES. ALL VALUES ARE REPORTED AS AN AVERAGE OVER THE TEST CASES IN CORRESPONDING CATEGORY (EASY, MEDIUM OR HARD).

Choices of Backbones	No. of Parameters	Easy		Medium		Hard	
		IoU-Score	F1-Score	IoU-Score	F1-Score	IoU-Score	F1-Score
MobileNetV2	∼8 millions	0.9423	0.9645	0.8873	0.9256	0.48210	0.53491
EfficientNetB0	∼10 millions	0.9499	0.9708	0.9063	0.9345	0.5721	0.6406
EfficientNetB3	∼17 millions	0.9603	0.9749	**0.9371**	0.9514	0.699	0.7649
EfficientNetB4	∼25 millions	**0.9773**	**0.9890**	0.9324	**0.9689**	**0.73451**	**0.79104**

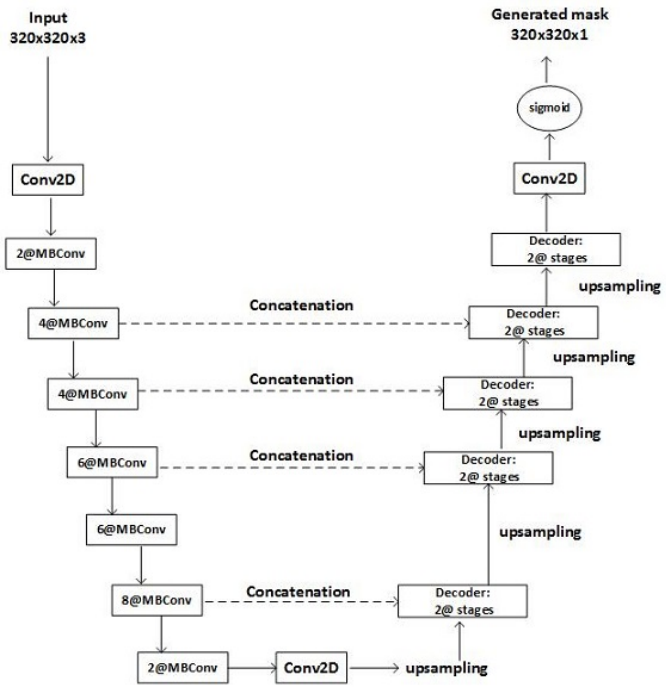

Fig. 3. FDL-Net Architecture: FDL-Net is a U-Net style network consists of two paths, contracting (or encoder) and expansive (or decoder). EfficientNetB4 is the backbone selected in encoder path. The main block used in this backbone is mobile inverted bottleneck, denoted by MBConv, along with squeeze-and-excitation optimizer. Each step in encoder path includes a number of MBConv blocks that shown before symbol '@'. Each Conv2D block consists of a conv2D layer followed by batch normalization and activation layers. In expansive path, there are couple of up-sampling steps each followed by a decoder block, each has two stages. Each stage includes a conv2D layer, followed by batch normalization and activation layers. The output of up-sampling step in expansive path, is concatenated with the corresponding feature map from the contracting path.

point cloud is approximated by finding the nearest points to the mapped center approximating the forged area. The mapping works for attacks with distinct segmentation mask to compute the center for mapping. In cases where the neural network fails to localize forgery, the mapping fails due to lack of distinct region. Sparsity of object and occlusions also impact the mapping. In hard cases representing farther objects, the sparsity is very high leading to loss of structure causing the mapping to fail. The hyperparameter K depends on the object and category of attack. We empirically average K to 300 for Easy, 200 for Medium and 100 for Hard cases on commonly forged objects such as pedestrians, cyclists, vehicles, and road signs covering the region of interest i.e., the immediate front of the forged object facing the concerned autonomous system.

VI. EXPERIMENTS AND PERFORMANCE

In this section, we summarize the performance of the FDL-Net model based on the evaluation metrics described in Section V-A. We also visualize the segmentation results on test cases by our trained FDL-Net model and mapping to 3D point cloud data. To demonstrate the superiority of the FDL-Net, we also train it with two benchmark datasets CASIA V1.0 and V2.0 and compare our model's performance with other state-of-the-art approaches.

All the training procedures are conducted on Ubuntu 18.04.1 using GPU GeForce GTX 1080 with 8 GB Memory and with CPU of Intel Core i7, 16 GB memory.

A. FDL-Net Training

We implemented FDL-Net using a Python library called Segmentation Models [36] based on Keras [37] and TensorFlow [38]. To initialize the weights, we use the pre-trained weights on 2012 ILSVRC ImageNet dataset [28] for each backbone. Our generated dataset is randomly divided with the ratio of 7:2:1 for training, validation and testing sets respectively. The input data (2D images and their ground truth masks) are normalized before training or evaluation phase. We also apply the transformations such as horizontal flip, Gaussian noise, brightness, contrast, and colors manipulations to augment the data on the fly. We used Adam as optimizer with learning rate initially set to 0.0001 and reduce it by factor of 0.1 once learning stagnates. The batch size is equal to 8 during training and validating.

We train FDL-Net for 80 epochs that is where our model training converges. We further fine tune the decoder by freezing the encoder layers. This fine tuning phase uses the well extracted features and starts updating the weights during the decoder re-training. Such a fine tuning helps to improve the prediction of segmented mask for Hard attacks and increases the model performance by 3.93% for Intersection over Union score and 3.16% for F-1 score.

TABLE II
MODEL PERFORMANCE ON CASIA V2.0 AND CASIA V1.0. ALL VALUES ARE REPORTED AS AN AVERAGE OVER THE TEST CASES.

CASIA V2.0				CASIA V1.0	
Test set 1		Test set 2		Test	
IoU-Score	F1-Score	IoU-Score	F1-Score	IoU-Score	F1-Score
0.823	0.871	0.812	0.859	0.598	0.641

B. Time Performance

As we discussed earlier, having a model with lower inference time to predict and segment the data is crucial in

TABLE III
F1 SCORE COMPARISON ON TWO DATASETS, CASIA V1.0 AND CASIA V2.0. ALL VALUES ARE REPORTED AS AN AVERAGE OVER THE TEST CASES. '-' DENOTES THAT THE RESULT IS NOT AVAILABLE IN THE LITERATURE.

Methods	CASIA V1.0	CASIA V2.0
RGB-N [25]	0.408	-
Edge-enhanced MFCN [23]	0.541	-
Patched-based method [39]	-	0.5634
U-Net [8]	-	0.749
RRU-Net [40]	-	0.841
FDL-Net (ours)	**0.641**	**0.859**

autonomous driving domain. FDL-Net model takes time of 66 milliseconds in average to detect and localize forged area in 2D RGB image. With more computation power, the time taken for forgery detection and localization decreases. Also localization of forgery in 3D point cloud data takes time around 230 milliseconds in average, run on MATLAB using a system with CPU of Intel Core i7, 3.30 GHz and 24.0 GB RAM. The time needed to localize forgery in 3D point cloud is higher due to matrix computations for point cloud data with dimensions over 23790×4 followed by a K-NN search over all points. Therefore, it's highly depends on the number of points in 3D data.

C. Model Comparison On CASIA Dataset

To demonstrate the superiority of the FDL-Net, we also train it with two benchmark datasets CASIA V2.0 that contains sufficient number of manipulated images (5123 tampered RGB images in the format of TIFF and JPEG), and CASIA V1.0 that contains 912 tampered images with size of 384x256, in the format of JPEG. Since their corresponding segmentation masks are not available, they are constructed based on the difference between tampered and original images. To train the FDL-Net on CASIA V2.0, We select those images with size of 384x256, then randomly divided into the train and validation sets (denoted as Test set 1 in Table II) and the rest of the images is considered as a test set (denoted as Test set 2 in Table II for evaluation purpose). The main reason for such a division is that since the CASIA V2.0 contains very elaborated fine-grained forgery with smoothed tampered contours, any pre-processing such as resizing or re-scaling can remove the forgery footprint and results in a model with poor performance. In evaluation phase, we apply window sliding with size of 384x256 over the test image to scan the entire of it and find the forged area if exists. The similarity of the reported results shown in Table II guarantees that such a train/val/test division works for this challenging dataset.

Table II also shows the performance of the model obtained by training on CASIA V1.0 based on the metrics of IoU and F1 scores. Furthermore, Table III summarizes the other models' performance in comparison with our approach that obtains promising results in comparison with other state-of-the-art approaches.

D. Visualization on Test Cases

(i) **CASIA V2.0 Test Cases**: First, we demonstrate the performance of the model, trained on CASIA V2.0 on multiple test cases that contain forged area in different sizes. FDL-Net can detect and localize the segmentation masks for unseen and fine-grained test data containing splicing forgery, the first four test cases as shown in Figure 4 and copy-move forgery (the last two test cases). In the latter type of forgery, the added object, selected from the same image, is re-scaled or rotated and placed into the scene.

Fig. 4. Qualitative results for splicing and copy-move forgery localization on CASIA V2.0 Dataset. a) RGB manipulated images, b) generated segmentation masks by FDL-Net, and c) ground truth masks.

(ii) **Forged KITTI Test Cases**: To evaluate our model for forgery localization on 2D and 3D data, we visualize the segmentation results in 2D RGB image and 3D point cloud data on Easy type attacked test case as shown in Figure 5. The image in leftmost side shows a pedestrian as a forged object in front of the autonomous vehicle, the rightmost side image displays the ground truth mask and middle one is the generated mask by FDL-Net model. It's worth noting that all the ground truth masks are displayed after applying binerization step so the forged region has sharp edges. Figure 6 shows the mapping result of 2D forgery localization to 3D point cloud data by the approach explained in Section V-C.

Figure 7 (Appendix A) depicts a category of Medium attack, in which the area containing a car, driving in left lane is the region under attack. FDL-Net model generates the segmentation mask, shown in middle image. In Figure 8 (Appendix A), we notice that the object semantics is lost to a great extent and

Fig. 5. 2D forgery localization for Easy category by FDL-Net model. a) Attacked RGB image, b) generated segmentation mask, c) binary ground truth segmentation mask.

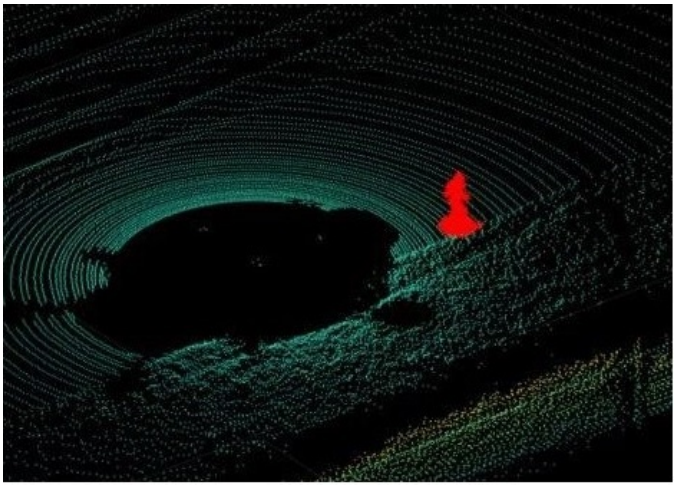

Fig. 6. 3D forgery localization of area under attack displayed in Figure 5. The LiDAR data is cropped to demonstrate the forged region clearly. In LiDAR data, the sensors installed on autonomous vehicles collect information in 360 degrees and compute the depth accordingly. To understand the LiDAR data, you can imagine the autonomous vehicle in the center of dark circle and the pedestrian (the forged area indicated in red) is walking in front of the car.

only a vague outline is visible owing to increase in sparsity with distance from the LiDAR imaging source. The K-NN still localizes the area of the forgery successfully which is not the case for Hard attack as shown in Figure 10 (Appendix A). The attack in 2D image, shown in Figure 9 (Appendix A) is accurately detected by FDL-Net and we can map the center of the forged area to 3D in a close range but K-NN fails due to sparse nature of the far away object.

We also observe that FDL-Net model is not able to detect and localize forged area in some test cases. For instance, Figure 11 (Appendix A) shows an attacked scenario belongs to Hard attack category, that FDL-Net model is not able to localize the area under attack due to the small forged area. Since it's too far from the autonomous vehicle the level of immediate risk associated with it is less.

VII. Conclusion

The need to detect and localize forgery such as multimedia data of different modalities in high risk domains autonomous vehicles has of recent motivated research for the same. In this work, we propose a Vision Based Measurement approach by introducing FDL-Net to not only detect but also localize forgery in multimedia data namely 2D RGB and 3D Point cloud in near real time of 66 milliseconds and around 200 milliseconds respectively. The newer architecture allows FDL-Net to perform better not only in Easy, Medium categories but also on Hard category attacks that are difficult even for naked eye. The segmented forged area as detected by FDL-Net is then mapped and localized to 3D point cloud. Despite studies being in a nascent stage on the later, we are able to accurately localize the area of attack in 3D accurately for Easy and Medium categories. In case of some Hard attacks, our approach misses the detection and hence, the localization of forged area. The reason is the very few number of forged pixels in such attacks. In future, we will explore techniques to address such Hard cases to be detected.

Acknowledgment

This material is based upon work supported by the US Army Research Office (ARO) Grant W911NF-17-1-0299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ARO.

References

[1] M. Khanafer and S. Shirmohammadi, "Applied ai in instrumentation and measurement: The deep learning revolution," *IEEE Instrumentation & Measurement Magazine*, vol. 23, no. 6, pp. 10–17, 2020.

[2] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: the rising trend of vision based measurement," *IEEE Instrumentation & Measurement Magazine*, vol. 17, no. 3, pp. 41–47, 2014.

[3] S. Abtahi, S. Shirmohammadi, B. Hariri, D. Laroche, and L. Martel, "A yawning measurement method using embedded smart cameras," in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2013, pp. 1605–1608.

[4] S. S. Beauchemin, M. A. Bauer, T. Kowsari, and J. Cho, "Portable and scalable vision-based vehicular instrumentation for the analysis of driver intentionality," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 391–401, 2011.

[5] H. Koch, A. Konig, A. Weigl-Seitz, K. Kleinmann, and J. Suchy, "Multisensor contour following with vision, force, and acceleration sensors for an industrial robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 2, pp. 268–280, 2012.

[6] K. D. Sharma, A. Chatterjee, and A. Rakshit, "A pso–lyapunov hybrid stable adaptive fuzzy tracking control approach for vision-based robot navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 7, pp. 1908–1914, 2012.

[7] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, "Measuring calorie and nutrition from food image," *IEEE Transactions on Instrumentation and Measurement*, vol. 8, no. 63, pp. 1947–1956, 2014.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[9] K. Bahirat, N. Vaishnav, S. Sukumaran, and B. Prabhakaran, "ADD-FAR: attacked driving dataset for forensics analysis and research," in *Proceedings of the 10th ACM Multimedia Systems Conference, MMSys 2019, Amherst, MA, USA, June 18-21, 2019*, M. Zink, L. Toni, and A. C. Begen, Eds. ACM, 2019, pp. 243–248. [Online]. Available: https://doi.org/10.1145/3304109.3325817

[10] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[11] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, 2013.

[12] K. Bahirat, U. Shah, A. A. Cardenas, and B. Prabhakaran, "Alert: Adding a secure layer in decision support for advanced driver assistance system (adas)," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1984–1992.

[13] Y. Zhan, Y. Chen, Q. Zhang, and X. Kang, "Image forensics based on transfer learning and convolutional neural network," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 165–170.

[14] B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, 2016, pp. 5–10.

[15] Y. Zhang, J. Goh, L. L. Win, and V. L. Thing, "Image region forgery detection: A deep learning approach." *SG-CRC*, vol. 2016, pp. 1–11, 2016.

[16] B. Chen, X. Qi, Y. Wang, Y. Zheng, H. J. Shim, and Y.-Q. Shi, "An improved splicing localization method by fully convolutional networks," *IEEE Access*, vol. 6, pp. 69 472–69 480, 2018.

[17] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[18] J. Ouyang, Y. Liu, and M. Liao, "Copy-move forgery detection based on deep learning," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2017, pp. 1–5.

[19] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2016, pp. 1–6.

[20] J.-S. Park, H.-G. Kim, D.-G. Kim, I.-J. Yu, and H.-K. Lee, "Paired mini-batch training: A new deep network training for image forensics and steganalysis," *Signal Processing: Image Communication*, vol. 67, pp. 132–139, 2018.

[21] B. Liu and C.-M. Pun, "Locating splicing forgery by fully convolutional networks and conditional random field," *Signal Processing: Image Communication*, vol. 66, pp. 103–112, 2018.

[22] X. Wang, H. Wang, S. Niu, and J. Zhang, "Detection and localization of image forgeries using improved mask regional convolutional neural network," *Mathematical biosciences and engineering: MBE*, vol. 16, no. 5, pp. 4581–4593, 2019.

[23] R. Salloum, Y. Ren, and C.-C. J. Kuo, "Image splicing localization using a multi-task fully convolutional network (mfcn)," *Journal of Visual Communication and Image Representation*, vol. 51, pp. 201–209, 2018.

[24] R. E. Yancey, "Bilinear faster rcnn with ela for image tampering detection." *CoRR, abs/1904.08484*, 2019.

[25] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[26] D. S. of California. (2018) Testing of autonomous vehicles with a driver. [Online]. Available: https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing

[27] J. Dong, W. Wang, and T. Tan, "Casia image tampering detection evaluation database," in *2013 IEEE China Summit and International Conference on Signal and Information Processing*. IEEE, 2013, pp. 422–426.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[29] W. Zhu, Y. Huang, L. Zeng, X. Chen, Y. Liu, Z. Qian, N. Du, W. Fan, and X. Xie, "Anatomynet: Deep learning for fast and fully automated whole-volume segmentation of head and neck anatomy," *Medical Physics*, vol. 46, no. 2, pp. 576–589, 2019. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.13300

[30] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html

[31] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 2815–2823.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[33] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[34] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[35] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[36] P. Yakubovskiy, "Segmentation models," https://github.com/qubvel/segmentation_models, 2019.

[37] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[39] P. Rota, E. Sangineto, V. Conotter, and C. Pramerdorfer, "Bad teacher or unruly student: Can deep learning say something in image forensics analysis?" in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2503–2508.

[40] X. Bi, Y. Wei, B. Xiao, and W. Li, "Rru-net: The ringed residual u-net for image splicing forgery detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

Appendix A
Results for Medium and Hard Attacks on KITTI Data Set

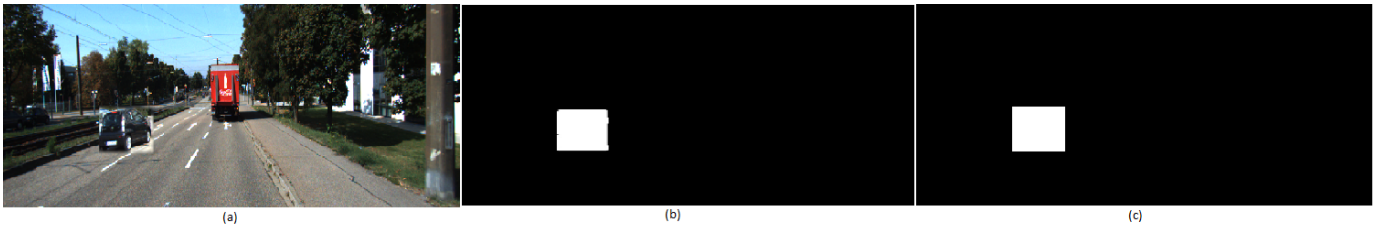

Fig. 7. 2D forgery localization for Medium category by FDL-Net model. a) Attacked RGB image, b) generated segmentation mask, The model detects and localizes the additive car into the scene very accurately, c) binary ground truth segmentation mask.

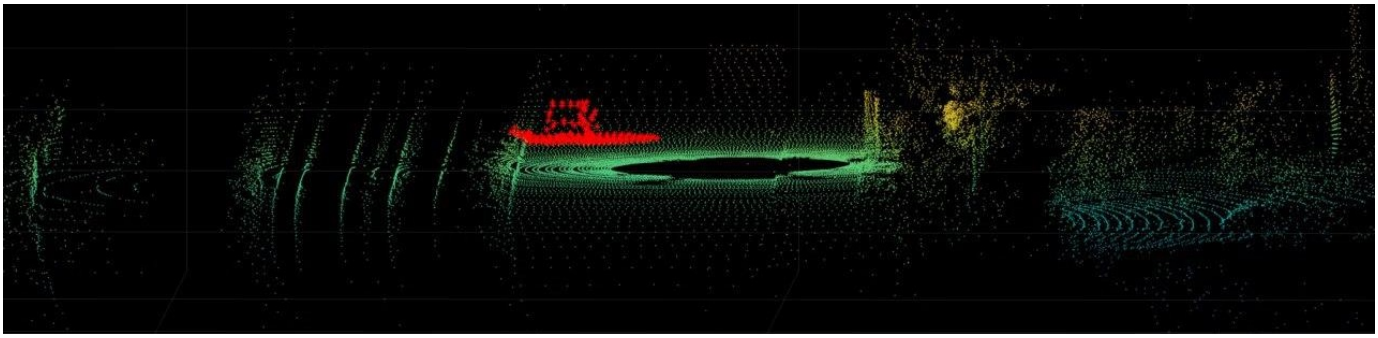

Fig. 8. 3D forgery localization of area under attack (in red) displayed in Figure 7

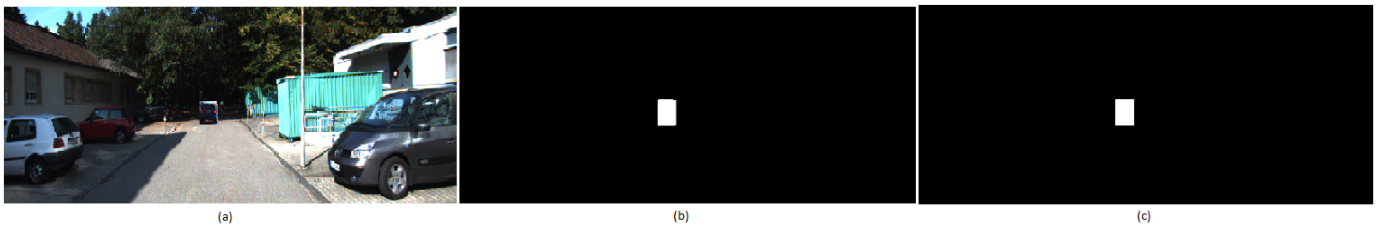

Fig. 9. 2D forgery localization for Hard category by FDL-Net model. a) Attacked RGB image, b) generated segmentation mask, c) ground truth segmentation mask.

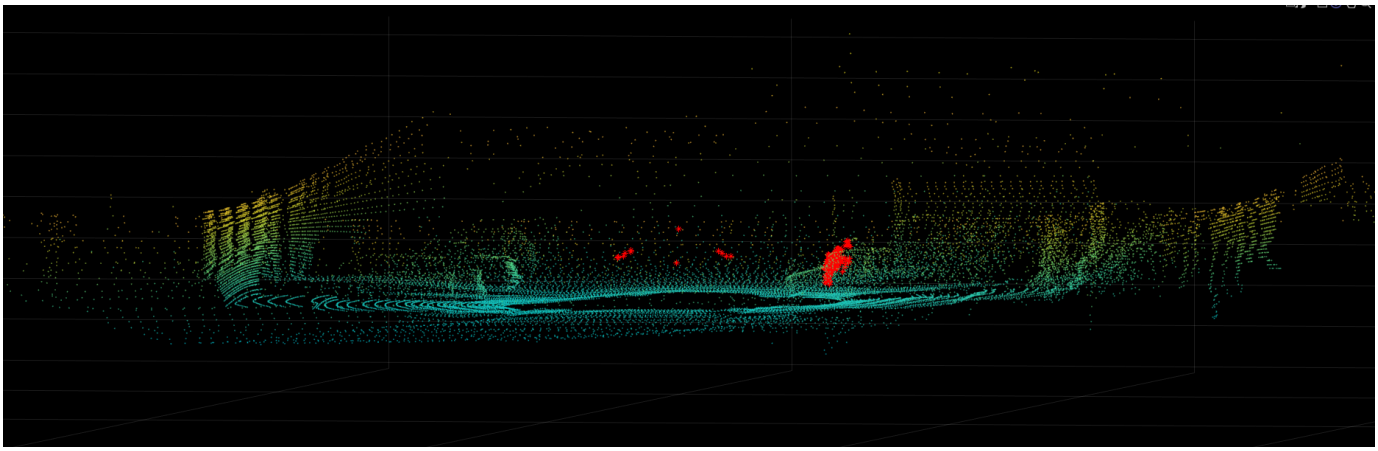

Fig. 10. 3D forgery localization of area under attack displayed in Figure 9. Although it can find the center point of attacked area properly K-NN fails to localize the forged area in this case.

Fig. 11. The failed test case of 2D forgery localization for Hard category. The forged area is a car located at the end of the road, indicated by a yellow bounding box. If the autonomous vehicle approaches to the additive object in next frames during the streaming the model can detect it as a forgery and set an alarm for proper reaction.