Lightweight Delivery Detection on Doorbell Cameras

Pirazh Khorramshahi[1]*, Zhe Wu[2], Tianchen Wang[2], Luke Deluccia[2] and Hongcheng Wang[3]
[1]Qualcomm Technologies, [2]Comcast Applied AI Research, [3]Amazon
pirazhkhorramshahi@gmail.com, {zhe_wu, tianchen_wang, luke_deLuccia}@comcast.com,
hongcheng.wang@gmail.com

Abstract

Despite recent advances in video-based action recognition and robust spatio-temporal modeling, most of the proposed approaches rely on the abundance of computational resources to afford running huge and computation-intensive convolutional or transformer-based neural networks to obtain satisfactory results. This limits the deployment of such models on edge devices with limited power and computing resources. In this work we investigate an important smart home application, video based delivery detection, and present a simple and lightweight pipeline for this task that can run on resource-constrained doorbell cameras. Our method relies on motion cues to generate a set of coarse activity proposals followed by their classification with a mobile-friendly 3DCNN network. To train we design a novel semi-supervised attention module that helps the network to learn robust spatio-temporal features and adopt an evidence-based optimization objective that allows for quantifying the uncertainty of predictions made by the network. Experimental results on our curated delivery dataset shows the significant effectiveness of our pipeline and highlights the benefits of our training phase novelties to achieve free and considerable inference-time performance gains.

1. Introduction

Computer Vision has become potent thanks to advances in Deep Learning to an extent that long standing problems like object detection, semantic segmentation, face and human action recognition can now be solved with high accuracy. Despite this, we have to highlight that this often comes at the price of significant computational burden which is typically accelerated by the use of Graphical Processing Units (GPU), Tensor Processing Units (TPU), or Neural Processing Units (NPU). Therefore, the degree to which computer vision tasks can be solved on edge devices with limited resources and computational power is constrained. Among these tasks is **Delivery Detection** which is concerned with recognizing delivery of merchandises (package, food, groceries, mail, etc.) at front doors to provide timely notifications for customers. Note that delivery detection task is different from package detection in that it identifies the instances of delivering items rather than the mere detection of packages which is currently practiced in smart home solutions. Delivery detection has numerous advantages including prevention of food perishing and porch piracy to name a few. According to the package theft annual report[1], in a twelve month period from 2021 to 2022, there has been more than 49 million package theft incidents in the United States alone with the estimated value of $2.4B. The prevalence of smart devices including smart doorbell and security cameras through out residential locations facilitates the development and adoption of automated delivery detection systems which can significantly reduce losses. Fig. 1 shows captured deliveries by doorbell cameras. Despite potential applications, delivery detection is a challenging task. Type, shape and size of packages can be quite diverse. Cardboard boxes in various size, mail, grocery bags, and food are among the items that are frequently delivered. Additionally, there are various courier services including United States Postal Services (USPS), United Parcel Services (UPS), DHL and Amazon, as well as growing number of smaller companies like DoorDash and Uber Eats especially after the COVID-19 pandemic. This translates to delivery personnel having diverse outfits and appearances as evidenced by Fig. 1. Finally the temporal extent of delivery events have high variance. Smaller

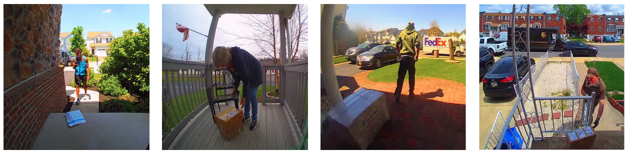


Figure 1. Sample delivery events captured by doorbell cameras.

*This work was done as the internship project at Comcast Applied AI Research.

[1] https://security.org/package-theft/annual-report/

items are delivered in a matter of seconds while delivering heavier objects can take much longer in the order of minutes. This also depends on submitting the proof of delivery in the form of a picture. Existing solutions such as Ring, Nest Hello, Arlo, AWS Rekognition, and Vivint mainly require cloud processing which results in higher bandwidth utilization, computation, and increased subscription fees. In addition, transferring data creates privacy concerns as opposed to local processing. Moreover, these methods primarily focus on detecting packages/boxes and not the instances of deliveries. In addition, package detection may fail as small or occluded packages are harder to be detected. Therefore, we set out to propose a solution to detect delivery instances that can be implemented on edge devices. We present a lightweight system that relies on motion detection to generate event proposals followed by their classification with a mobile-friendly 3DCNN [17] network. Through the novel incorporation of an attention mechanism and benefiting from the the theory of evidence and subjective logic, we significantly improve the base performance of this system without imposing additional processing costs. In summary, this paper makes the following contributions:

- We introduce a lightweight delivery detection system running on doorbell cameras with ARM Cortex-A family of processors. In contrast to the widely used package detection in the industry, our system is to detect the delivery events from videos.
- We propose a semi-supervised attention module in 3DCNNs to extract robust spatio-temporal features.
- We propose to adopt evidential learning objective to quantify the uncertainty of predictions and enforce a minimum certainty score to ensure quality predictions.

2. Related Work

Development of CNNs has substantially contributed to the remarkable improvements in the status of video action recognition. [38] presented a two-stream design to leverage both RGB and Optical Flow modalities to capture spatio-temporal cues. With the prevalence of 3D convolutional kernels [17], I3D network was introduced to better model temporal interactions and established a strong baseline [5]. Authors in [12] proposed a two-step approach to localize potential activities from hierarchical clustering of detected objects and recognize a wide range of activities in surveillance cameras from Optical Flow modality using a modified I3D, namely TRI3D, to adjust the temporal bounds of localized activities. Despite strong performance, I3D incurs high computational cost due to its depth and significant number of 3D filters. To reduce the computational burden, authors in [45] proposed S3D network in which only deeper layers of the network are designed to capture temporal information, namely top-heavy design. Additionally they propose to factorize 3D convolutional filters into spatial and temporal layers to reduce computational complexity. This is also proposed by [42] in their R(2+1)D network design to improve the efficiency of action recognition models. In another line of work [10] proposed a two-path design to capture spatial semantics at a reduced frame rate and temporal cues at finer resolution in a faster and lighter pathway; the design known as slow-fast, has improved the accuracy/efficiency trade-off significantly. With the introduction of transformers [43], many works soon adopted transformers in the context of video understanding. [11] uses the base of I3D network to obtain initial spatio-temporal features upon which proposals are generated and corresponding features are passed to a stack of action transformer units. To improve efficiency, [9] proposed a multi-scale network MViT to generate multi-scale feature hierarchies by spatio-temporal down-sampling as well as down-sampling of the dimensionality in attention heads. Similarly, [29] adopted SWIN transformer [28] for video action recognition to reduce the quadratic complexity of standard transformer modules. Authors in [1] designed ViViT with factorized encoding scheme to ingest tokenized input sequences and lower the complexity associated with the running of transformer blocks. Despite these progress, these models are heavyweight which limits their deployment on a device with limited power and compute budget [22]. There have been attempts to leverage transformer-based architectures for mobile applications; however, they are mainly limited to 2D vision applications [33] as scaling these models to 3D and temporal modeling is non-trivial. Other attempts to adopt transformers for human action recognition in a mobile environment, use non-visual modalities such as inertial measures [8] which cannot be used for spatially fixed cameras *e.g.* surveillance cameras. Therefore, in this work, we present a lightweight CNN-based pipeline that can run on edge devices and consistently improve its performance by devising a novel attention module and benefiting from the the theory of evidence and subjective logic in the training objective.

3. Method

To the best of our knowledge, there are no published research on the delivery detection task. Therefore, we first establish a simple, intuitive, and easy to implement baseline. Next, we propose our novel delivery detection system to process untrimmed videos which overcomes the shortcomings of the baseline.

3.1. Baseline System

Our baseline model which is shown in Fig. 2, is a two-stage method where the first stage is responsible to identify whether a person is present in the scene for a given frame.

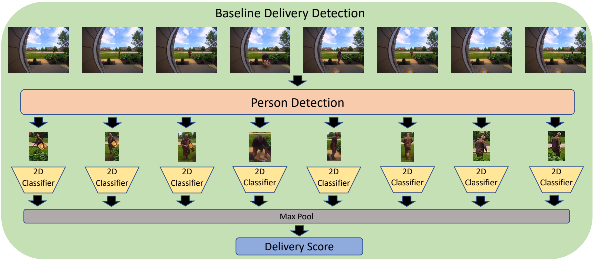


Figure 2. Baseline delivery detection pipeline. In each frame, a person detector localizes a person with highest detection confidence and passes the person's crop to a 2D classifier to obtain a delivery score. Delivery scores from all the sampled frames in the video snippet are max-pooled to generate the final delivery score.

In case a person is detected, second stage crops the person from the full frame, and passes it to a 2D classifier to generate a delivery score $s_i \in [0, 1]$ (i is the frame index) where larger s_i indicates higher chance of a delivery personnel. The system queries the scene at a fixed rate of 1 frame per second over the continuous intervals of length 15 seconds. Therefore, each chunk is summarized by 15 delivery scores; max-pooling is used to obtain the final delivery score, *i.e.* $s = \max_{i=1}^{15} s_i$. Backbone architectures for person detector and 2D classifier are MobileNet-SSD [27] and EfficientNet-B0 [40] respectively to meet the constraints imposed by the resource-limited hardware of a doorbell camera. Adoption of max-pooling facilitates the implementation and results in an efficient system which can be easily interpreted. However, this cultivates a high False Positive Rate (FPR). In case, the second-stage outputs a relatively high delivery score for a single frame due to an artifact, or sudden variation in illumination, the system makes a false detection despite opposing evidence from other frames. This will be discussed in more details in section 4. Moreover, this design requires two individual modules which prevents the end-to-end optimization of the overall system and the temporal information that can provide critical cues delivery recognition is not considered. This motivates us to devise a system that can model temporal interactions and is optimized in an end-to-end fashion.

3.2. Proposed Delivery Detection System

As mentioned above, rich temporal information that is critical to detect deliveries, are not captured by the baseline as frames are processed individually. For instance, a person getting close to a residence, bending towards the ground, and moving away is a strong indication for a delivery event. Therefore, having a model that accounts for temporal interactions across neighboring frames, creates the opportunity to learn enhanced representations. To this end, we propose to use a lightweight 3DCNN model that can process multiple frames at a time and extract rich temporal semantics.

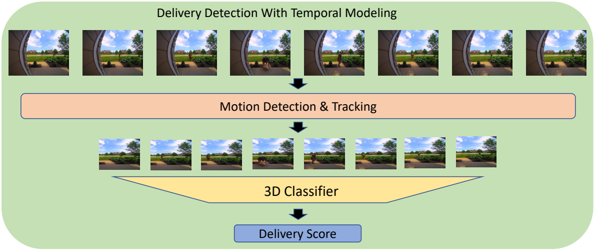
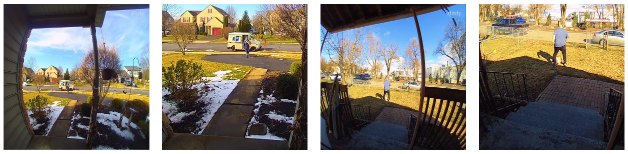

Delivery Detection With Temporal Modeling
Motion Detection & Tracking
3D Classifier
Delivery Score

Figure 3. Proposed delivery detection pipeline. Motion algorithm detects and tracks foreground motion blobs. The foreground motion is used to reduce the spatial extent of frames to motion regions. Once certain number of frames are gathered, they are passed to a 3D classifier to obtain the delivery score.



(a) Full frame (b) Cropped (c) Full frame (d) Cropped

Figure 4. Using motion to generate tighter activity proposals. (a) and (b) are full and cropped frames of a delivery event. (c) and (d) are full and cropped frames of a non-delivery event.

Fig. 3 shows the overview of the proposed pipeline. For smart doorbell cameras, we need to have a mobile-friendly design that can be accommodated by the limited computational budget. To achieve this, we propose to use motion detection and tracking algorithm to reduce the spatial extent of frames to where motion occurs followed by a classifier to differentiate delivery from non-delivery events. Below, we discuss motion detection and tracking algorithm. Next, due to the computation-intensive nature of 3D convolutional and transformer based networks, we first discuss the lightweight networks that are mobile friendly. Based on this we set a 3D baseline and then design a novel semi-supervised attention module and adopt an evidential optimization objective to improve performance while preserving the computational complexity.

3.2.1 Motion Detection & Tracking

To generate spatially-tighter proposals for delivery events compared to using full frames, we propose to use foreground motion to focus on regions that activities happen. This enables us to preserve a better pixel resolution as lightweight networks more often than not require small spatial size, *e.g.* 112x112. In case of using frames in their entirety, activity regions can only occupy a few number of pixels. Fig. 4 shows the impact of using motion to generate tighter action proposals. Therefore, motion detection is an important pre-processing step to generate activity proposals. The algorithm for motion detection is based on Mixture of Gaussians (MOG) for background/foreground segmentation which adaptively models each pixel by a mixture of Gaussian distributions. This generates foreground motion mask containing motion blobs that are refined via adopting connected components. When a motion blob passes two thresholds, namely active time and variance, a motion event is triggered to signal the camera to query the scene. Active time indicates period of time in which a motion blob is continuously detected and tracked based on centroid distance measure. Variance criteria shows how much a blob has moved in the camera's field of view. This helps removing waving flags, leaves, and swaying trees which generate trivial motion events. Once a motion event is triggered, a thumbnail of fixed size is placed on the region where motion blobs reside.

3.2.2 Lightweight Backbone Architectures

As discussed in section 2, 3D transformers are not particularly suited for mobile platforms to process visual data, hence we only focus on CNN based networks. Compared to their 2D counterparts, 3DCNNs have the ability to learn temporal interactions. This is achieved via additional parameters and higher computational complexity. Therefore, their adoption in resource-limited applications is constrained. To address this, [23] introduced 3D versions of MobileNetv1 [15], MobileNetv2 [36], ShuffleNetv1 [46], ShuffleNetv2 [32], and SqueezeNet [16] which were developed for 2D mobile applications. Moreover, these networks are pre-trained on the Kinetics [19], a large-scale human action recognition dataset, to provide robust weight initialization in the context of transfer learning when used in downstream tasks. We use these networks as our candidate backbone architectures. We performed initial experiments on a subset of our curated dataset discussed in section 4. Fig. 5 compares the performance of these networks to distinguish delivery from non-delivery events in terms of Precision-Recall (PR) curve and F_1 score. Since MobileNetv2 obtains the highest accuracy, we base all our subsequent analysis on this network.

3.2.3 Semi-supervised Attention

We are interested to investigate opportunities of improving accuracy without introducing additional compute and run-time complexity. Inspired by the works of [7, 20, 35] which incorporate the paradigm of curriculum learning [3] for object detection and vehicle analytics tasks, we devise a training mechanism to simplify the learning of delivery versus non-delivery events at early stages of optimization and gradually make the task more realistic as training progresses. The motivation for this simplification is that people can provide critical cues to distinguish deliveries. During training, we explicitly excite parts of the intermediate feature maps of the network that correspond to the location of people in the scene so that the network can better extract such signatures. However, as training progresses we gradually reduce our assistance, *i.e.* the degree to which excitation happens, to let the network learn to performs enhanced feature extraction on its own. Consequently, this method of training can be viewed as a semi-supervised approach. Once training is concluded, excitation stops and the computational complexity will be the same as the original 3D MobileNetv2. To excite the output of the l^{th} layer f_l of shape $C * T * H * W$ where C, T, H, and W represent number of channels, frames, height and width respectively, we generate T binary single-channel masks of shape $H * W$ denoted by m_l in which pixels corresponding to the bounding box location of people are set to one while the rest are set to zero. Afterwards, channel-averaged feature maps $\tilde{f}_l = \sum_{c=1}^{C} f_l(c, ., ., .)/C$ are multiplied with m_l in a point-wise manner. The resulting tensor is then multiplied by a scalar $\alpha[n]$ which is a function of training epoch n as follows: $\alpha[n] = 0.5 * (1 + \cos(\pi n/N))$ where N is the total number of epochs. The ensuing excitation component is finally added to the input feature maps. Fig. 6 demonstrates the excitation operation and Equation 1 expresses the mathematical relationship between the excited f_l^e and original f_l feature maps.

$$f_l^e = f_l + \alpha[n] * (\tilde{f}_l.m_l) \quad (1)$$

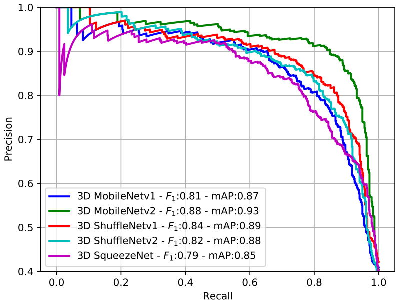


Figure 5. Precision-Recall comparison of lightweight 3DCNN architectures on the test set of our Doorbell delivery detection dataset discussed in section 4.1.

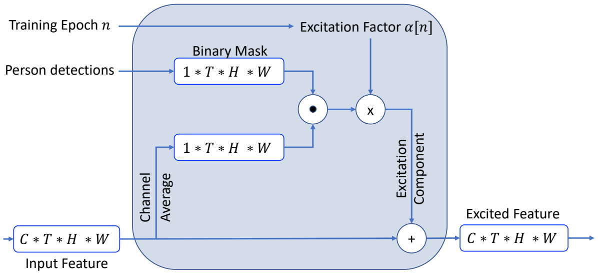


Figure 6. Excitation layer

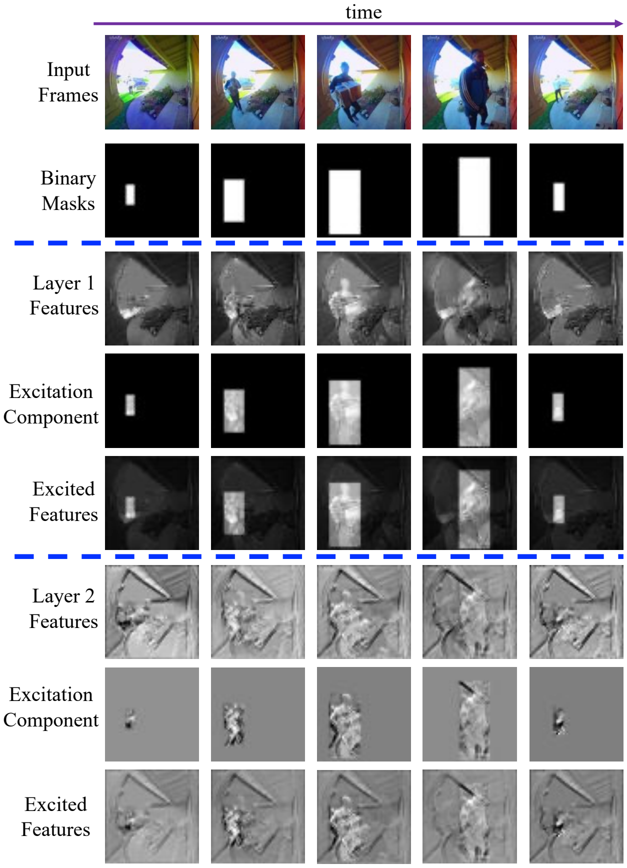


Figure 7. Exciting the first and second layer features of 3D MobileNetv2 for a sample video snippet in the first training epoch.

For 3D MobileNetv2, we restricted the excitation to the outputs of first and second convolutional layers since they have the same temporal resolution as the input and the content of feature maps are not as abstract as deeper layers of the network. The impact of such excitation on the first and second layer features of the 3D MobileNetv2 is shown in Fig. 7. Note that how regions of the feature map containing a person are highlighted with respect to the rest. This helps the network to focus more on the visual information of people and extract more robust representations in early stages of training.

3.2.4 Evidence-based Delivery Detection

To differentiate delivery from non-delivery events, a straightforward learning objective would be to obtain logits corresponding to each of the two classes followed by maximizing the likelihood of each input sample $p(y|x, \theta)$ where x, y, and θ are input sample, corresponding label, and model parameters. In practice this is achieved via Cross-Entropy loss which applies softmax function and minimizes the negative log-likelihood of the true class. While this approach is widely used, it that does not account for uncertainties when making predictions and only provides point estimates of class probabilities. In addition, the relative comparison of class probabilities cannot be used to quantify prediction uncertainty as softmax is known to inflate the probabilities [13]. In contrast [37] proposed a learning objective based on the theory of evidence [6] and subjective logic [18] in which a predictor is tasked to gather evidence for any of the possible outcomes to formulate classification in conjunction with uncertainty modeling by considering a Dirichlet distribution as a prior on class probabilities. To realize this, an evidence function g (which can be implemented as either ReLU, exponential, or soft-plus) is applied to the output of the network h to ensure that outputs are always non-negative and are representative of the amount of evidence gathered by the network for each of the K classes:

$$e_i = g(h_i(x; \theta)), \quad i = 1 \dots K \tag{2}$$

where x and K are the input video and the number of classes respectively. This is equivalent to gathering $K + 1$ mass values u and $b_i, i = 1 \dots K$ which are related through $u + \sum_{i=1}^{K} b_i = 1$. u is the uncertainty of the prediction and b_i is the belief mass corresponding to the i^{th} class which is related to the evidence of i^{th} class via $b_i = e_i/S$ where $S = \sum_{i=1}^{K}(\alpha_i)$ is referred to as the total strength of the Dirichlet distribution and $\alpha_i = e_i + 1, i = 1 \dots K$ are Dirichlet parameters. Based on this, uncertainty can be written as $u = K/S$ which is inversely proportional to the total strength S or the total evidence gathered by the network $\sum_{i=1}^{K} e_i$. Therefore, gathering high evidence results in small uncertainty and vice-versa. Since class probabilities are assumed to follow Dirichlet distribution, *i.e.* $\mathbf{p} \sim \text{Dir}(\mathbf{p}|\alpha)$ where $\mathbf{p} \in \mathbb{R}^K$, the average probability of the i^{th} class can be computed as α_i/S [37]. Therefore, the resulting loss function is computed via the following formulation:

$$\mathcal{L} = -\sum_{i=1}^{K} \mathbf{y}_i(\log(\alpha_i) - \log(S)) \tag{3}$$

$\mathbf{y}_i$ is the i^{th} entry of the one-hot encoding label vector. Despite providing quantifying the uncertainty of predictions, this approach is deterministic and may suffer from the overfitting caused by the training of neural networks. To alleviate this issue, a number of regularization terms are proposed. [37] proposed a Kullback-Leibler (KL) term to encourage the network to generate zero evidence for a sample if it cannot be correctly classified. This is achieved by removing the generated evidence for the true class, *i.e.* $\tilde{\alpha}_i = \mathbf{y}_i + (1 - \mathbf{y}_i).(e_i + 1)$, and minimizing the KL distance of the corresponding Dirichlet distribution $\text{Dir}(\mathbf{p}|\tilde{\alpha}_i)$ from the one with zero total evidence, *i.e.* $S = K$ and $\text{Dir}(\mathbf{p}|\mathbf{1})$, which represents a uniform distribution. Note that $\mathbf{1}$ is the notation for all one vector. This KL term essentially discourages the network to over-fit and to generate evidence for samples about which it is uncertain:

$$\begin{aligned}\mathcal{L}_{KL} = &\log(\frac{\Gamma(\sum_{i=1}^{K} \tilde{\alpha}_i)}{\Gamma(K)\prod_{i=1}^{K}\Gamma(\tilde{\alpha}_i)}) \\ &+ \sum_{i=1}^{K}(\tilde{\alpha}_i - 1)\left(\psi(\tilde{\alpha}_i) - \psi(\sum_{j=1}^{K}\tilde{\alpha}_j)\right)\end{aligned} \tag{4}$$

where $\Gamma(.)$ and $\psi(.)$ are gamma and logarithmic derivative of gamma function respectively. In addition to $\mathcal{L}_{KL}$ regularization, [2] propose to calibrate the feature extraction network to be confident for its accurate predictions while being uncertain for it false predictions. To realize this goal, authors propose to maximize the Accuracy versus Uncertainty (AvU) utility function defined in [24]. AvU is formally defined as:

$$\text{AvU} = \frac{n_{AC} + n_{IU}}{n_{AC} + n_{AU} + n_{IC} + n_{IU}} \tag{5}$$

In Eq. 5, n_{AC}, n_{AU}, n_{IC}, and n_{IU} are number of accurate and confident predictions, number of accurate and uncertain predictions, number of inaccurate and confident predictions, and number of inaccurate and uncertain predictions respectively. A well calibrated model should obtain high n_{AC} and n_{IU} and low n_{IC} and n_{AU}. To regularize the learning of the model to achieve this objective, we draw inspiration from [2] and add the following calibration objective to the overall loss function.

$$\begin{aligned}\mathcal{L}_{cal} = &- \lambda_n \mathbb{1}(\tilde{y} = y) p \log(1-u) \\ &- (1-\lambda_n)\mathbb{1}(\tilde{y} \neq y)(1-p)\log(u)\end{aligned} \tag{6}$$

where $\hat{y} = \arg\max_i\{\alpha_i/S\}$, and $p = \max_i(\alpha_i/S)$ for $i \in \{1 \dots K\}$ are the predicted class label and predicted probability for a given input sample. Note that $\mathbb{1}(.)$ is the indicator function. Moreover, λ_n is an epoch-dependent weight to adjust the contribution for each of the terms in the right hand side of Eq. 6. Specifically, $\lambda_n = \lambda_0 e^{-\ln(\lambda_0)n/N}$ is set to be exponentially-increasing ($\lambda_0 < 1$) with respect to epoch index n. The intuition is that over the initial training epochs the model mainly makes inaccurate predictions and therefore it should be penalized to increase uncertainty for these predictions via $\mathbb{1}(\tilde{y} \neq y)(1-p)\log(u)$. However, as training progresses the model makes accurate predictions more often and therefore it should reduce the corresponding uncertainties which is enforced by $\mathbb{1}(\tilde{y} = y)p\log(1-u)$.

4. Experiments

Here we first describe the dataset we gathered and curated for our experiments. Afterwards the implementation details will be discussed followed by the presentation of the experimental results.

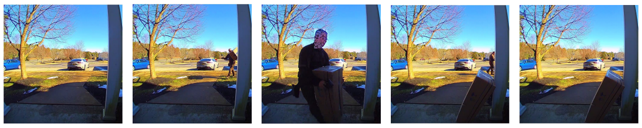

(a) 0 s (b) 82 s (c) 90 s (d) 94 s (e) 144 s

Figure 8. Extent of deliveries are shorter than the length of videos. Here, in a 144-seconds long video, delivery only lasts 12 seconds.

4.1. Dataset

To the best of our knowledge, there are no publicly available dataset that is suited for our application. The closest dataset is UCF-Crime dataset [39] which includes only a handful of videos from security cameras installed at residential areas which cover a wide range of anomaly events that are not of interest for our study. Therefore, we choose to collect a video dataset by recording video snippets from static doorbell cameras installed at the front door of 339 residences whose residents approved and signed the designated data collection agreement for this purpose. To ensure all videos contain activities, we started to record only when there was a motion trigger and stopped recording around two minutes after the start. This resulted in 5477 videos with the resolution of 1280x960. Sample frames of this dataset are shown in Fig. 1. In the initial round of annotation process, all the videos were assigned a video-level tag to denote whether at least a delivery event happens during the entire duration of the video, which resulted in 1898 delivery and 3579 non-delivery video samples. However, we note that delivery events only occupy a small portion of the video as shown in Fig. 8. Therefore, to train our 3DCNN we require finer annotation for the start and end time of an instance within the video. Given a delivery event involves at least one person, to collect finer information, we run person detection and tracking on all videos to obtain person tracks. Afterwards we annotated person tracks with delivery/non-delivery tags. We used EfficientDet-D4 [41] for detection and DeepSort [4, 44] multi-object tracker with embeddings extracted by the bag of tricks for person re-identification [31] model with ResNet50_IBN [34] backbone implemented in FastReID [14] to compute tracks. This led to the collection of 2057 person tracks delivering items and 2930 person tracks that do not correspond to any delivery events, *e.g.*, entering or exiting the residence, and playing in front lawn. Despite gathering fine annotations with tight spatial and temporal bounds, our delivery detection system is intended to be deployed on proposals generated based on motion events that are not tight around people in time and space and do not necessarily involve people, *e.g.*, passing vehicles, presence of pets or wildlife. Therefore, we need to generate and label motion events to prepare our training and testing data. To generate these events we use

Table 1. Doorbell delivery detection dataset statistics. Note that cameras across splits are disjoint.

Split	# Cameras	# Delivery Videos / Events	# non-Delivery Videos / Events
Train	182	1016 / 2324	1902 / 3817
Validation	59	416 / 595	769 / 1680
Test	98	466 / 706	900 / 1751
Total	339	1898 / 3625	3579 / 7248

the algorithm outlined in section 3.2.1. To assign labels, labeled person tracks of delivery events are used to accelerate the process. To this end, we measured the overlap between a computed motion event and all person tracks within a video in terms of temporal and spatial Intersection over Union (IoU):

$$IoU_t = \frac{T(m_i \cap t_j)}{T(m_i \cup t_j)}, \qquad IoU_s = \frac{A(m_i \cap t_j)}{A(m_i \cup t_j)} \tag{7}$$

where m_i, t_j are i^{th} motion event and j^{th} person track. In addition, $T(.)$, $A(.)$ are temporal length and the spatial area functions respectively. Note that initially we tried to use spatio-temporal (3D) IoU for measuring the overlap for a motion-track pair; however, we noticed that the overlap values become quite small even for closely matched pairs due to measuring the volume. Using two different thresholds, namely temporal t_{min} and spatial s_{min}, provides more flexibility. Accordingly, 3625 positive and 7248 negative motion events were gathered and split into train, validation and test sets as reported by Table 1.

4.2. Implementation Details

Each motion proposal is uniformly sampled by 16 frames which are spatially resized to 112x112 by preserving the original aspect ratio. During training, temporal jittering of ± 10 frames is applied to the start and end bounds of proposals as a form of data augmentation. Color jittering is also employed with the probability of 0.2; brightness, contrast and saturation factors are uniformly chosen from $[0.9, 1.1]$ while hue factor is uniformly selected from $[-0.1, 0.1]$. Additionally, generated cuboids are horizontally flipped with the probability of 0.5. To optimize Adam [21] with decoupled weight decay AdamW [30] is employed with gradient L_2 norm clipping at 0.25 and the learning rate that is linearly warmed up to $5e-4$ in the first 5 epochs and decayed at 20^{th} and 40^{th} epochs with gamma factor of 0.1. Weight decay is set to $5e-4$ and the network is trained for the total of 50 epochs. To alleviate the impact of easy samples, focal loss [25] with focusing parameter of 1.0 is adopted.

4.3. Evaluation Metric

As our target task is to classify an input video as delivery or non-delivery, we use F_1 score and mean Average Precision (mAP) that presents the area under the precision-recall curve (AUC). Note that for a given video, multiple motion events may occur. In case the tag of the video is delivery, the classifier must classify at least one of those events as delivery and if the tag is non-delivery, the classifier must classify all the events as non-delivery. As we noticed high number of false positives with 2D baseline system 3.1, we report the FPR as well.

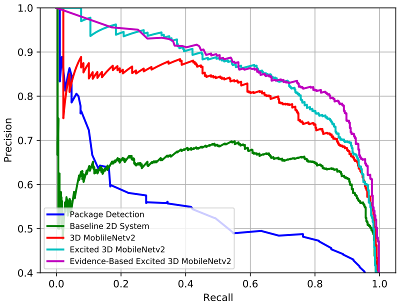


Figure 9. Precision-Recall comparison of package detection, baseline 2D pipeline and variants of 3D MobileNetv2 on the test set.

Table 2. Performance comparison for detecting delivery events on the test of Doorbell Delivery Detection dataset.

	Evaluation Metrics			
Model	F_1(↑)	mAP(↑)	FPR(↓)	Classification Accuracy (%)(↑)
Package Detection	0.59	0.54	0.42	63.74
2D Baseline	0.73	0.64	0.24	79.75
3D MobileNetv2	0.77	0.80	0.19	83.02
Excited 3D MobileNetv2	0.79	0.85	0.15	85.05
Evidence-based Excited 3D MobileNetv2	**0.81**	**0.86**	**0.13**	**86.11**

4.4. Experimental Results

This section compares the 2D baseline system described in section 3.1, the 3D MobileNetv2, the 3D MobileNetv2 with semi-supervised attention module outlined in section 3.2.3, namely excited MobileNetv2, and the excited 3D MobileNetv2 optimized with the evidence-based objective of section 3.2.4. Additionally, as package detection is offered in many smart home solutions, we choose to evaluate its applicability for delivery detection task. To this end, we train a COCO [26] pre-trained MobileNet-SSD v2 object detector on 4405 manually labeled images with package annotations. Fig. 9 plots the precision-recall curves and Table 2 reports evaluation metrics for each of these models when evaluated on the test set. Unsurprisingly, package detection has a significantly inferior performance as detecting small and occluded packages is challenging. Also drastic variation in size, shape and appearance of delivered items

Table 3. Efficiency comparison between baseline 2D system and the proposed system based on 3D MobileNetv2 architecture.

System	Input	Inference time (ms/input)	Binary Size (MB)	FLOPS (G)
2D Baseline	1x1280x960	44.5	7.0	1.41
3D MobileNetv2	16x112x112	69.94	2.9	0.55

further exacerbates this performance gap. Therefore, solutions based on package detection are not suited for detecting delivery instances. From Table 2 it is also seen that a 2D model compared to 3D alternatives performs at a lower level due to inability of modeling temporal interactions. We observe the increase of 4% and 16% in F_1 and mAP scores when we incorporate the temporal information via 3D MobileNetv2 which also results in reducing the FPR by 5 points generating much fewer false delivery notifications. Additionally, training the 3D MobileNetv2 with our novel semi-supervised attention module and the incorporation of evidence-based optimization objectives increases F_1 and mAP scores by 5.1% and 7.5% and reduced the FPR by 31%. These novelties further enhances the performance of 3D MobileNet in a meaningful manner without introducing any additional overhead during test time. This is of great importance for a resource limited design to maintain a fixed computational budget during inference.

It is also important to compare the 2D baseline model with the 3D MobileNetv2 in terms of run-time speed, binary size of quantized models and the number of floating point operations (FLOPS) which are critical when deployed on an edge device. Note that Excited 3D MobileNetv2 and Evidence-based Excited 3D MobileNetv2 share the same run-time, binary size and FLOPS as 3D MobileNetv2 during inference. Table 3 provides this comparison. While the inference time of 2D baseline system is 36% lower, we have to note that the system continuously queries the scene at a fixed rate and performs 2.5 times more operations (1.41 GFLOPS) per forward pass compared to 3D MobileNetv2 (0.55 GFLOPS) which performs inference once a motion event is concluded. Also the required memory to store 3D MobileNetv2 is much smaller compared to the 2D baseline which provides opportunities for increasing the complexity and potentially the accuracy of a prospective model. Finally we would like to highlight an additional benefit of using an evidence-based objective compared to Cross-Entropy. We can compute the average uncertainty score for the samples within the validation set on which the model made mistakes. We further use this value as a threshold to remove the predictions whose uncertainty values are higher when processing the test set as presented in Table 4. By applying this threshold, we remove 89 videos from our test set which not only increases F_1 and mAP scores, but also reduces the FPR. Here we visualize randomly selected samples about which the model was uncertain in Fig. 10. It is seen that, these sample have flavors of delivery events. For instance, in (a) a mailman is going towards a neighboring house, in (b) a family member is holding a tablet which is what most delivery personnel do after delivering an item to submit proof of delivery, in (c) a mail man is picking an item from front door for either shipping or returning, in (d) a food delivery person with no uniform is seen, and in (e) the resident is putting down her belongings at the front door. Therefore, uncertainty score can be used effectively to reduce the number of false predictions.

Table 4. Removing samples with uncertainty score above 0.16 computed from the validation set. Total samples removed from test set: 89 videos.

	Evaluation Metrics			
Model	$F_1(\uparrow)$	mAP($\uparrow$)	FPR($\downarrow$)	Classification Accuracy (%)($\uparrow$)
Evidence-based Excited 3D MobileNetv2	0.81	0.86	0.13	86.11
Evidence-based Excited 3D MobileNetv2 + Uncertainty Removal	**0.83**	**0.87**	**0.12**	**87.92**

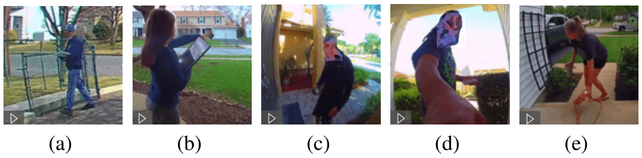

(a) (b) (c) (d) (e)

Figure 10. Uncertain predictions by our network.

5. Conclusion

In this work we presented a mobile-friendly pipeline to perform the task of delivery detection in contrast to package detection on resource-limited platforms such as doorbell cameras. The proposed system has the capacity of modeling temporal interactions in video streams to enhance its predictions over a 2D model. In addition, we have improved the accuracy of our designed system by a considerably through the novel incorporation of a semi-supervised attention module, namely excitation layer. We have also benefited from the advances in the theory of evidence and subjective logic to modify the optimization objective of the system. This not only boosts the system performance but also quantifies the uncertainty of predictions made by the network and provides the opportunity to enforce a minimum level of certainty to further advance predictions. We emphasize that all these improvements are achieved without adding any inference-time computation and memory overhead to the proposed design.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 2

[2] Wentao Bao, Qi Yu, and Yu Kong. Evidential deep learning for open set action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13349–13358, 2021. 6

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009. 4

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 6

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2

[6] Arthur P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(2):205–232, 1968. 5

[7] Mohammad Mahdi Derakhshani, Saeed Masoudnia, Amir Hossein Shaker, Omid Mersa, Mohammad Amin Sadeghi, Mohammad Rastegari, and Babak N Araabi. Assisted excitation of activations: A learning technique to improve object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9201–9210, 2019. 4

[8] Sannara EK, François Portet, and Philippe Lalanda. Lightweight transformers for human activity recognition on mobile devices. *arXiv preprint arXiv:2209.11750*, 2022. 2

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[11] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019. 2

[12] Joshua Gleason, Rajeev Ranjan, Steven Schwarcz, Carlos Castillo, Jun-Cheng Chen, and Rama Chellappa. A proposal-based solution to spatio-temporal action detection in untrimmed videos. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 141–150. IEEE, 2019. 2

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 5

[14] Lingxiao He, Xingyu Liao, Wu Liu, Xinchen Liu, Peng Cheng, and Tao Mei. Fastreid: A pytorch toolbox for general instance re-identification. *arXiv preprint arXiv:2006.02631*, 2020. 6

[15] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 4

[16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 4

[17] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. 2

[18] Audun Jøsang. *Subjective logic*, volume 3. Springer, 2016. 5

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[20] Pirazh Khorramshahi, Sai Saketh Rambhatla, and Rama Chellappa. Towards accurate visual and natural language-based vehicle retrieval systems. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4183–4192, June 2021. 4

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[22] Raivo Koot, Markus Hennerbichler, and Haiping Lu. Evaluating transformers for lightweight action recognition. *arXiv preprint arXiv:2111.09641*, 2021. 2

[23] Okan Kopuklu, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. Resource efficient 3d convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 4

[24] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33:18237–18248, 2020. 6

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 7

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 7

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C

Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[29] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 2

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[31] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 6

[32] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 4

[33] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *arXiv preprint arXiv:2110.02178*, 2021. 2

[34] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 464–479, 2018. 6

[35] Neehar Peri, Pirazh Khorramshahi, Sai Saketh Rambhatla, Vineet Shenoy, Saumya Rawat, Jun-Cheng Chen, and Rama Chellappa. Towards real-time systems for vehicle re-identification, multi-camera tracking, and anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 622–623, 2020. 4

[36] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 4

[37] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. 5

[38] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2

[39] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018. 6

[40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 3

[41] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 6

[42] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[44] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649, 2017. 6

[45] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 2

[46] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. 4