Scene Invariant Virtual Gates using DNNs

Simon Denman*, Clinton Fookes*, Prasad K.D.V. Yarlagadda†, Sridha Sridharan*
*Image and Video Laboratory
†School of Chemistry, Physics and Mechanical Engineering
Queensland University of Technology (QUT), Brisbane, Australia
Email: {s.denman, c.fookes, y.prasad, s.sridharan}@qut.edu.au

***Abstract*—Understanding where people are located and how they are moving about an environment is critical for operators of large public spaces such as shopping centres, and large public infrastructure such as airports. Automated analysis of CCTV footage is increasingly being used to address this need through techniques that can count crowd sizes, estimate their density, and estimate the through-put of people into and/or out of a choke-point. A limitation of using CCTV based approaches however is the need to train models specific to each view, which for large environments with 100's or 1000's of cameras, can quickly become problematic. While some success has been had in developing scene invariant crowd counting and crowd density estimation approaches, much less attention has been given to developing scene invariant solutions for through-put estimation. In this paper, we investigate the use of CNN and LSTM architectures to estimate pedestrian through-put from arbitrary CCTV viewpoints. To properly develop and demonstrate our approach, we present a new 22 view database featuring 44 hours of pedestrian throughput annotation, containing over $11,000$ annotated people; and using this proposed approach we show that we are able to outperform a scene dependant approach across a diverse set of challenging view-points.**

I. Introduction

Despite the prevalence of CCTV cameras in most public places, there still exists a large number of pedestrian monitoring solutions that rely on either fixed, specialist cameras, or other sensor technologies (i.e. infra-red laser, bluetooth). In particular, the task of counting people as they pass through a point (i.e. a doorway) is a common problem encountered in a number of environments for which a suite of specialist hardware solutions exist. Furthermore, such technologies are often used in areas already well covered by CCTV, raising the possibility of using CCTV feeds to directly address the counting problem and avoiding the need for additional hardware expenditure.

While a number of approaches have been proposed to estimate pedestrian throughput from CCTV imagery, they have typically had one principal drawback: they are scene dependant, i.e. a model needs to trained for each view that we wish to obtain a count for. While for isolated installations this may not be too problematic, for large public infrastructure such as airports, where there are potentially 100's of sites to be monitored, the need to perform training and/or calibration for each individual view or gate can make such approaches impractical and difficult to scale to large installations. The problem is further complicated by the diverse nature of the areas to be monitored, with both large and small check-points, located indoor and outdoor, potentially being of interest in the one facility.

In this paper we present an approach to achieve this task in a scene invariant manner using convolutional and recurrent neural networks. Through the use of CNNs and LSTMs, we are able to achieve a level of scene invariance, making the approach far more practical for large-scale deployment in real world environments. We investigate the use of both grey scale and optical flow data in isolation and in tandem, and explore how LSTMs can be leveraged to exploit the sequential nature of the problem to improve performance. In this work, we also introduce a new dataset that contains 44 hours (2 hours for 22 different gate locations) of pedestrian footage taken from a variety of indoor and outdoor locations, which includes over $11,000$ pedestrian annotations. The remainder of this paper is organised as follows: Section II discusses prior work on pedestrian throughput estimation; Section III outlines our proposed approach; Section IV presents the proposed database and evaluation protocol; Section V demonstrates the performance of the proposed networks on our database; and Section VI concludes the paper.

II. Prior Work

Early approaches to the problem of pedestrian throughput estimation sought to use overhead cameras [1], [2], [3], [4], [5], [6], [7], from which people can be easily located and counted through techniques such as motion segmentation. However while the use of highly constrained camera systems such as these do offer a degree of site invariance (i.e. if all sites have the same or very similar overhead view, then deployment across multiple sites is simplified), their deployment requires a significant investment in new infrastructure, as the vast majority of existing CCTV cameras will not have a suitable field of view.

As such, more recent research has sought to develop methods that can work with arbitrarily placed cameras, by extracting features over a line, or from within a region of an image. Kim et al. [8] proposed the concept of the 'virtual gate' for counting crowds past a point. Kim et al [8] uses a single line in the image, and observed optical flow perpendicular to the line over time. The observed flow is integrated and scaled by a learned coefficient to obtain a count. Similar approaches have been proposed by [9], who introduced a fixed-length sliding temporal window, generating a larger set of samples to train a Bayesian Poisson regression model; and [10] who

used the concept of the influx and outflux count from a region of interest to count people as they passed through a region by tracking pixels on the boundary of the ROI. A region based approach proposed by Denman et al. [11] extracted aligned optical flow within a region of interest and used Gaussian Process Regression to count people within a small temporal window. In this case, the optical flow field is multiplied by a vector that describes the target direction such that motion in the target direction becomes positive, motion in the opposite direction becomes negative, and motion perpendicular to the direction of interest is ignored.

A limitation of these techniques is their scene dependence, i.e. a model needs to be trained for each view point. While this is not a significant problem for small sites with only a small number of cameras or areas to be monitored, for a large environment such an airport, with 100's of doors or passage ways that need to be monitored, such approaches rapidly become impractical.

Within the related area of crowd counting (i.e. counting everyone in a scene), much research has focussed on developing methods allow for scene invariance, such that models trained on one scene can be applied to another. Approaches such as the use of camera calibration to normalise feature vectors [12]; the use of Bayesian model adaptation [13] to transfer models learnt in one domain to a second; and the application of deep neural networks to learn invariant features [14] have all shown some success.

Recently, Cao et al. [15] and Zhao et al. [16] have both proposed the use of deep convolutional neural networks (DCNNs) for estimating crowd throughput. Cao et al. [15] combined three DCNNs based on AlexNet [17] to estimate the pedestrian flow from both visual and optical flow imagery, and the crowd state (indicating whether people are entering, leaving, both, or the scene is empty); and then combined these outputs into an estimated count. Zhao et al. [16] proposed training a network to solve the related tasks of crowd density estimation and crowd velocity estimation, and used the output of these processes to estimate the number of people crossing a line of interest. Both approaches in [15] and [16] are shown to offer good performance, including the ability to generalise between similar scenes (i.e. different cameras, but all with broadly front-on views of the region of interest). However, both of these approaches only utilise optical flow or estimated crowd velocity which is restricted to consecutive estimations. Both approaches ignore the inherent temporal aspect of the problem. As such, in this paper we investigate how recurrent neural networks, and in particular LSTMs [18] can be used to help estimate crowd throughput in a scene invariant manner.

III. Proposed Approach

We propose using a deep convolution neural network (DCNN) to count people as they pass a line in an image. We investigate three different DCNN architectures:

1) A 2D convolutional approach which takes striped grey-scale and optical flow images as input;
2) A naive LSTM approach, which takes the same striped image as above as input (i.e. one row per time-step); and
3) A 2D convolution with LSTM approach, which applies 2D convolutions to a sequence of images, which are fed into an LSTM to estimate crowd count over a time window.

As input to the networks, we consider the use of grey scale or optical flow images on their own, or in combination. Section III-A outlines the process of generating our input images, while Section III-B details the network architectures used in this paper.

A. Image Representation

1) 2D Representation: The proposed approach aims to count people as they pass a line defined in the image. As such, a simple 2D spatio-temporal representation can be generated by stacking slices of the image taken over a line of interest, such that

$$F(x,t) = I_t(L(x)), \tag{1}$$

where F is the feature image extracted for a given time window; t is the current time-step, such that the tth row in the output feature F is given by the tth image in the sequence, I_t; and $L(x)$ is a function that maps the x coordinate in the output feature to the line of interest in the input image. In this manner, a single image captures activity occurring over a small time window. The input images, I_t, can be either grey scale images, or aligned optical flow images. A number of examples for both grey scale images and aligned optical flow images are shown in Figure 1.

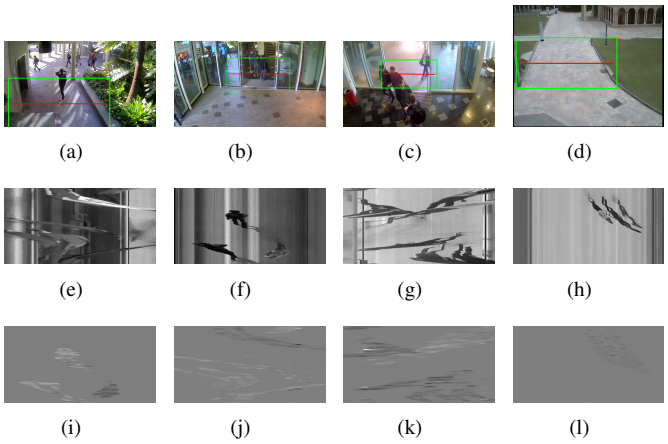


Fig. 1. Example Sliced Input Images: The top row shows a sample image from a scene, with the line of interest drawn in red. The second row shows example striped grey scale images taken over the line for a 10 second window, and the bottom row shows striped aligned optical flow images for the same time region and time window. The striped images are generated by stacking lines extracted from consecutive images, as outlined in Equation 1. The aligned optical flow images used in generating images on the bottom row and generated using Equation 2, followed by Equation 1, with the aim being to normalise the optical flow to cope with arbitrary orientations.

When considering optical flow, we need to normalise for the direction of interest to ensure consistent input to the network. We use the idea of aligned optical flow from [11] to ensure that the flow field is normalised such that positive values indicate motion in the direction of interest, negative values indicate movement in the opposite direction, and any motion orthogonal to the direction of interest is suppressed.

The optical flow field at time t is denoted v_t, and the optical flow at a pixel $\mathbf{p}$ is denoted $v_t(\mathbf{p})$. The component of this flow which points in the direction of interest $\mathbf{d}$ is referred to as the aligned optical flow, and is computed using the dot product,

$$\hat{v}_t(\mathbf{p}) = v_t(\mathbf{p}) \cdot \mathbf{d}. \tag{2}$$

In this manner, we can generate an aligned optical flow image, from which we extract and stack the line of interest, generating the images shown in Figure 1. For all scenes used in this paper, we re-sample the line of interest to length 100, and stack 50 consecutive frames captured at 5 frames per second such that the extracted images are 100×50 pixels in size and represents a 10 second time window.

2) 3D Representation: For networks that require a 3D input, an alternate representation to that presented in Section III-A1 is used. Rather than extracting a 2D line from each image and stacking these to create an image, we extract a 2D patch from each image, and stack these to create a volume,

$$F_{3D}(x, y, t) = I_t(R(x, y)), \tag{3}$$

where F_{3D} is the feature volume extracted for a given time window; t is the current time-step, such that the tth slice in the output feature F_{3D} is given by the tth image in the sequence, I_t; and $R(x, y)$ is a function that maps the x, y coordinate in the output feature to the region of interest in the input image.

As with the 2D representation, a grey-scale volume can be extracted by simply taking the raw pixel values, while an optical flow representation can be obtained by using the aligned optical flow approach described in Section III-A1. following the 2D approach, volumes are extracted such that the region of interest is 100×50 pixels in size centred around the line of interest, with 50 frames captured at 5 frames per second used such that the volume is $100 \times 50 \times 50$ pixels in size.

B. Network Architectures

We investigate both the use of a 2D convolutional network where a time window is represented as a 2D image composed of slices; and LSTM networks that aim to model the sequence and thus consider multiple observations in sequence over the time window. The two approaches are discussed in subsections III-B1 and III-B2 respectively.

All networks have a final output of size 2, and are trained to estimate the number of people passing through the line/region of interest for a fixed period of time in each direction. The first element of the final layer indicates the number of people passing in the forward direction, while the second indicates the number of people passing in the backwards direction. A secondary output, a pair of 12×6 pixel images such that one image is produced for each direction, is also produced by the networks. These images are trained to indicate the approximate region that each person who is counted in the input sequence occupies, and the output size is set to the rounded size of the input image down-sampled by a factor of 8 (i.e. after three 2×2 max pooling operations).

1) 2D Convolution Network: We use a 2D convolutional network which is made of a number of smaller convolutional units. Following the work of [19], we use a number of stacked smaller convolutional filters as opposed to a smaller number of larger convolutional filters. The basic convolutional unit of our network is presented in Figure 2 (a), and consists of two convolutional layers followed by a max-pooling layer. Batch normalisation is used to improve convergence [20], and we also use a 2D spatial drop-off [21] after the max-pooling layer, as we find that this further improves performance.

We stack three of the convolutional units as shown in Figure 3 (a). A convolutional filter of size 5×5 is used for the first group, and a filter of size 3×3 is used for the second two convolutional groups. 16, 32 and 64 filters are used in the groups respectively. We find that for this task, the smaller number of filters is sufficient (compared to other tasks such as image classification [19]) and using additional filters (or further convolutional layers) has little to no extra benefit. We follow the convolutional layers with three fully connected layers of size 256, 64 and 2 respectively (with batch normalisation and activation, see Figure 3 (b)). The final fully connected layer has non-negative constraints placed on the weights and biases, ensuring that the outputs of the network must be positive. As with the convolutional layers, both batch normalisation and drop out are used as we find this improved performance. The secondary output is generated using a fully connected layer that takes input from the last of the convolution groups, and maps the convolution output to a 144 length vector.

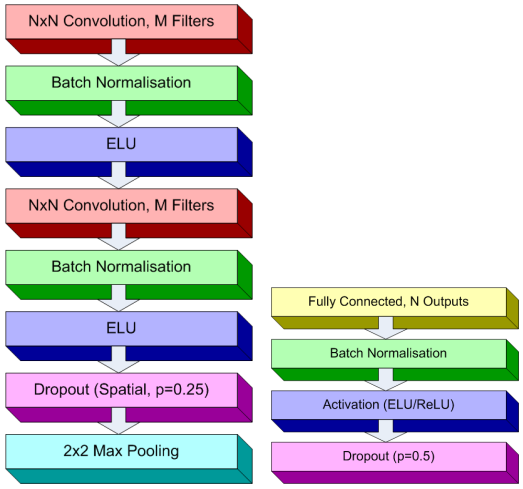
Within the network, we predominately use ELUs [22] as the activation function to improve learning, with the exception of the final activation, which is a ReLU. We revert to a ReLU as the final activation to ensure that only positive inputs are provided to the final fully connected layer that computes the counts, which (combined with non-negative weights and biases) ensures that the output pedestrian count estimates are greater than or equal to 0.

As can be seen in Figure 3, we investigate the use of both a single mode of input data (i.e. grey scale values or optical flow) and the use of two modes (grey scale and optical flow). When using the two modes as shown in Figure 3 (b), we use a group of three convolutional units for each mode. The flattened output of these convolutional units are merged prior to the fully connected layers, which are responsible for fusing the combined data to estimate the person counts.

2) LSTM Network: The task of pedestrian throughput estimation is inherently sequential, and thus well suited to recurrent neural networks such as LSTMs [18]. We investigate two ways to employ LSTMs:

1) Using the 2D representation of Section III-A1, where each image row now represents a single time step for the LSTM;
2) Using the 3D representation of Section III-A2, where a 2D image is obtained for each time step, which is first passed through a number of convolutional layers before being passed to an LSTM.

The first approach is illustrated in Figure 4 (b) and (c) for the single and dual mode networks respectively. The single mode



(a) Convolution Group (b) Fully Connected Group

Fig. 2. Network Components: (a) shows the convolution group that is used to build the two networks, consisting of two small convolution filters followed by a max-pooling; (b) shows the fully connected group, which consists of a fully connected layer, followed by batch normalisation, activation and a drop-out.

network (Figure 4 (b)) uses two stacked LSTMs and merely passes the input through the LSTMs (with the first being a sequence-to-sequence LSTM) followed by a fully connected layer (output size, 2) to get the throughput estimates. The dual mode approach (Figure 4 (c)) uses a sequence-to-sequence LSTM for each mode, and then combines these using a third LSTM (sequence-to-one) followed by a fully connected layer. As with the networks outlined in Section III-B1, constraints are placed on the final fully connected layer (positive weights and biases) to ensure positive outputs for the estimated throughputs. The secondary output is generated by connecting the output of the first LSTM (or first pair in the case of the dual input network) with a fully connected layer to generate an output vector of length 144.

The second approach is shown in Figure 5, and can be seen as the union of the 2D convolution approach in Section III-B1 and the simple LSTM approaches described above. Input images are passed through a network of 6 convolutional layers arranged in pairs of two (see Figure 3 (a) and (b)) to generate a sequence of deep features. This deep feature sequence is then passed through the LSTM and fully connected layers to generate the output estimates. The secondary output is derived from the output of the convolutional layers (as per the 2D approaches of Section III-B1), and the final fully connected layer has positive constraints on the weight and bias as per the other proposed networks.

C. Network Training and Loss

We seek to estimate the pedestrian throughput for an environment over a large period of time. As such, minimising bias and ensuring that the network can accurately estimate when no one is passing through the gate are important and should be considered by the loss function. For this reason, we propose the sum of the mean absolute error (MAE) and the mean squared error (MSE) as the loss function for the primary network output (the estimated counts),

$$L = \frac{1}{N}\sum |E(x) - GT(x)|) + \frac{1}{N}\sum (E(x) - GT(x))^2, \quad (4)$$

where $E(x)$ are the estimates and $GT(x)$ are the ground truth values; and N is the batch size. The first component, the mean absolute error (MAE), promotes the correct estimation of 0 values. The mean squared error (MSE) on the other hand promotes the correct identification of large groups moving through. We find that using only a single one of these terms leads to less accurate estimates, as networks trained with MAE alone have a tendency to underestimate large groups; while networks trained used MSE are prone to a small positive bias (i.e. estimating 0 as 0.1) which over times leads to large errors accumulating.

The secondary output (the coarse location map) is evaluated using the mean squared error. For this output we are less concerned with the presence of small values in place of 0's as we are seeking to use this input to assist with the detection of large groups; rather than as a direct output of the network. Weights for the loss functions are set to 1.0 and 0.2 for the counts and location map respectively.

In training the networks, we use a batch size of 32, and use the Nesterov Adam optimizer [23] with the parameters provided in the paper. We train all the networks for 30 epochs, and select the iteration that yields the best loss on the validation set.

IV. Data and Evaluation Protocol

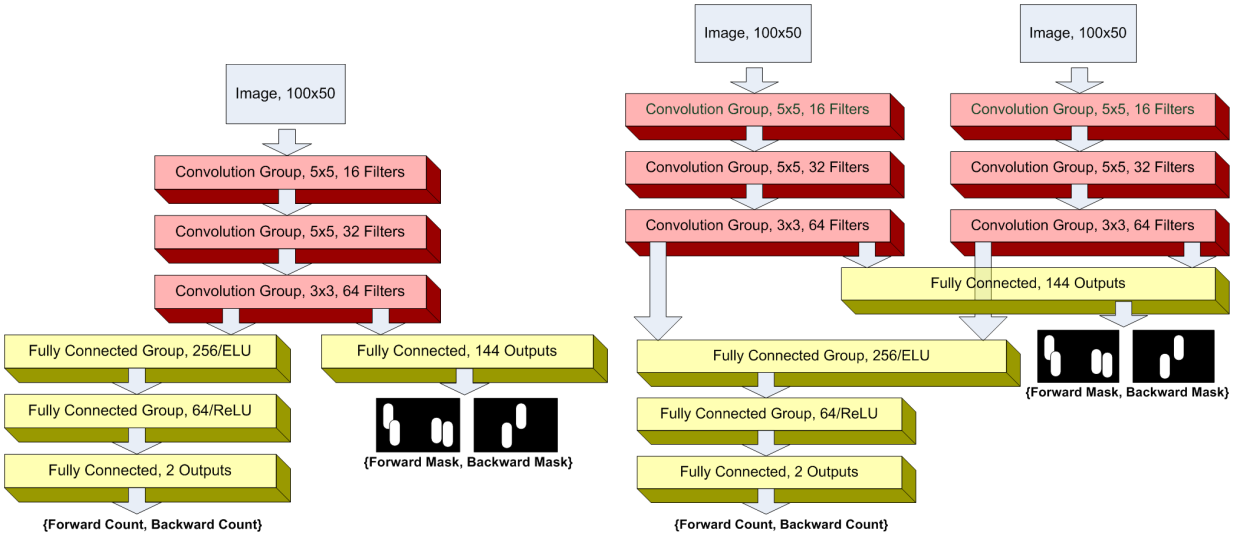
A. Data

An extended set of the data first presented in [11] is used, including 16 new two hour sequences. Data is collected for 22 locations across 14 cameras on a busy university campus, as shown in Figure 6. The gates are placed in a variety of locations and orientations, covering outdoor and indoor locations, doorways, elevators foyers, and pathways. The gates themselves are of varying real-world width, and some perspective distortion is present (particularly for those monitoring elevator foyers, see Figure 6 (r), (s) and (t)).

For each camera, two hours of continuous data is collected. To annotate pedestrian throughput, we simply annotate every instance of a person crossing the line of interest in either direction. A count is registered once the approximate centre of mass of the person crosses the line of interest [1]. Further details on the database, including the number of pedestrians in each view, are given in Table I and Figure 7.

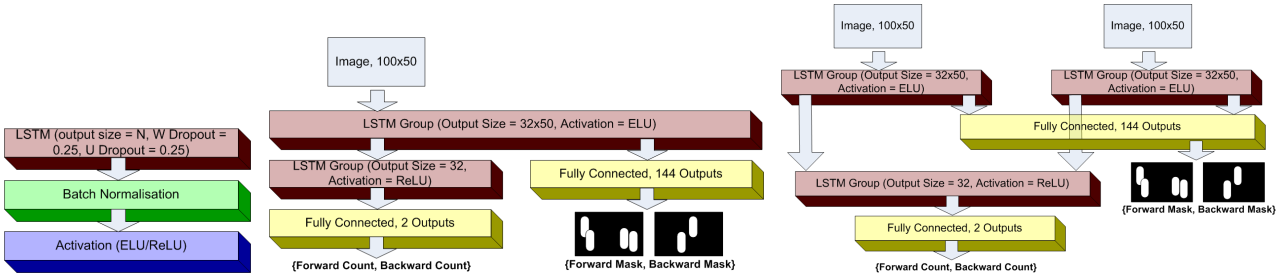
We note that the data contains a large number of windows with zero people crossing in either direction, which can bias the network training as it can become common for input batches to consist of only windows that contain no people. To reduce the number of instances of this, once at least 5 consecutive windows that contain 0 people are observed within a sequence, all subsequent consecutive windows that contain 0 people are removed from the training set (i.e. we remove

[1]To obtain a copy of the data, please contact the authors



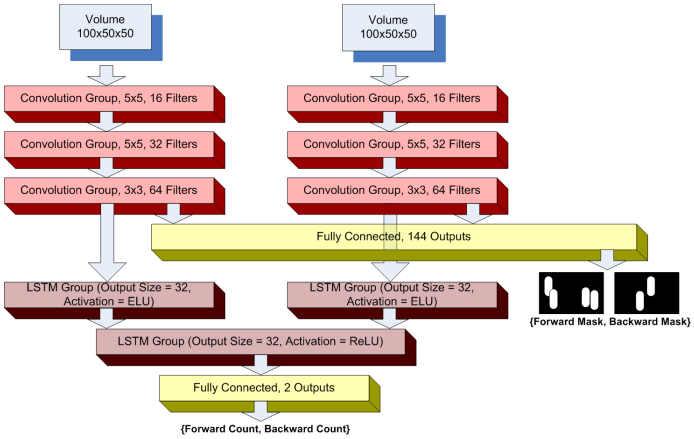
(a) 2D Single Mode Network (b) 2D Dual Mode Network

Fig. 3. 2D CNNs: The single and dual mode networks shown in (a) and (b) respectively both use three convolutional groups per mode followed by three fully connected groups. For the dual mode network, the first fully connected group receives the merged output of both the optical flow and grey scale convolution sub-networks.



(a) LSTM Group (b) Single Mode LSTM (c) Single Mode LSTM

Fig. 4. LSTM Network Architectures: (b) and (c) show networks that take the 2D representation and pass it through an LSTM, such that each slice of the input image becomes an observation. Both networks use stacked LSTMs, with the first LSTM outputting a sequence, and the second a single vector. This approach is taken to allow the secondary output to be generated. Each LSTM group is an LSTM layer, followed by batch normalisation and an activation as shown in (a).



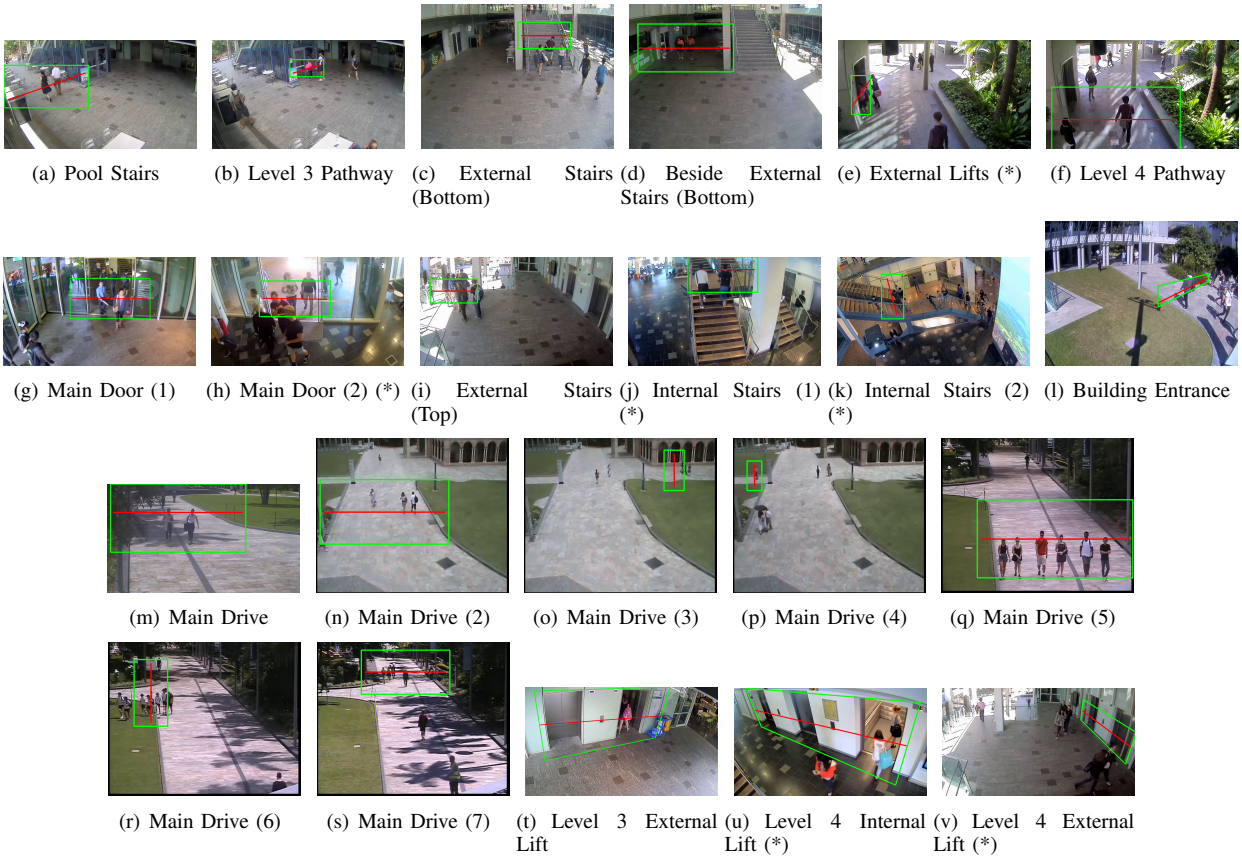
(a) 2D Convolution with LSTM

Fig. 5. 2D-LSTM Network Architectures: the network takes a 3D volume of data and extracts a convolutional features for each time step, which are subsequently passed into the LSTM as a sequence. As with the LSTM networks, stacked LSTMs are used, with the first layer of LSTMs generating a sequence. The intermediate output is generated based on the combined output of the convolutional layers.

all windows until a person appears again). While this does reduce the number of samples present for learning (see Table I), we find that we still have a sufficient number and ultimately achieve a better overall result due to the increased variation in the training data.

To estimate pedestrian flow, data is extracted as detailed in Section III-A, such that we have 10 second windows at 5 frames per second; and the line of interest (or width of the region of interest) is re-sampled to 100 pixels (roughly the minimum length of the 22 gates annotated in the video feeds). Colour images are converted to grey scale and normalised into the range $[-0.5, 0.5]$; while optical flow images are clipped at ± 50 (optical flow magnitudes of this this size and larger are very rare within the data, and when present represent an error in optical flow estimation) and are also normalised in to the range $[-0.5, 0.5]$. To obtain additional data for training, we use a sliding window sampling every 2 seconds, allowing us to extract a sufficient amount of data to train the networks.

The secondary output is generated from the ground truth such that a corresponding 100×50 pixel image mask indicating approximate people locations is created for each window. The



(a) Pool Stairs (b) Level 3 Pathway (c) External Stairs (Bottom) (d) Beside External Stairs (Bottom) (e) External Lifts (*) (f) Level 4 Pathway

(g) Main Door (1) (h) Main Door (2) (*) (i) External Stairs (Top) (j) Internal Stairs (1) (*) (k) Internal Stairs (2) (*) (l) Building Entrance

(m) Main Drive (n) Main Drive (2) (o) Main Drive (3) (p) Main Drive (4) (q) Main Drive (5)

(r) Main Drive (6) (s) Main Drive (7) (t) Level 3 External Lift (u) Level 4 Internal Lift (*) (v) Level 4 External Lift (*)

Fig. 6. The 22 gates taken from 14 cameras used in our evaluation. The red lines indicate the lines over which we collect the input images for the 2D convolutional network and LSTM network; and the green boxes indicate the regions that we extract for the 2D convolutional LSTM network. A (*) indicates that the data was originally available in [11].

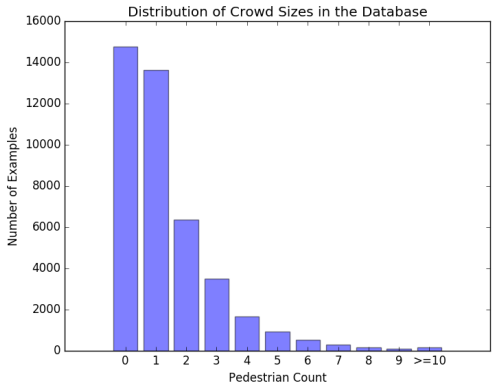


Fig. 7. Distribution of crowd densities in the proposed database, with long sequences of consecutive 0's removed.

location that the subject crosses the gate (line of interest) and the frame in which they are annotated are mapped into the range $[0..99, 0..49]$ to provide an image coordinate that indicates the person's centre in the generated mask. A rectangular template that represents the approximate duration that a person takes to traverse the gate (2.5 seconds) and size (10% of the gate width) is added to the mask at the corresponding location. Two masks are generated for each window: a mask for the forward direction and a mask for the reverse direction. Some example masks are shown in Figure 8, and full details of the process to generate the masks (as well as python source code) are provided with the dataset.

B. Evaluation Protocol

A leave one out cross validation approach is used to evaluate the system. For each fold we train on 17 gates, use 4 gates as a validation set, and test on the 22nd unseen gate. [2] Results are reported in terms of mean squared error (MSE) and mean absolute error (MAE) for the 10 second windows; and relative error (RE) over the entire sequence. We avoid using RE over sub-windows as for all gates, there are periods of time when there are no pedestrians present.

V. Results

We evaluate the proposed approach on the dataset presented in Section IV-A. The proposed networks are evaluated in Section V-A; and we analyse the impact of the loss function on estimation accuracy in Section V-B; assess the impact of

[2]Validation and test gates are approximately evenly spaced within the entire set of 22 gates, i.e. when gate 2 is the test gate, gates 1, 8, 13 and 18 are used for validation. This approach is taken to try and ensure that the validation set captures a reasonable cross-section of the conditions present. Note that gates are ordered as per Figure 6 and Table I, such that 'Pool Stairs' is gate 1.

Camera	Total People		Total Samples	
	Forward	Backwards	All Data	Reduced 0's
Pool Stairs	74	52	3,596	854
Level 3 Pathway	82	63	3,596	1,095
External Stairs (Bottom)	162	202	3,596	1,772
Beside External Stairs (Bottom)	211	156	3,596	2,045
External Lift	25	19	3,597	355
Level 4 Pathway	238	203	3,597	1,902
Main Door (1)	976	853	3,596	3,462
Main Door (2)	856	980	3,597	3,461
External Stairs (Top)	388	425	3,600	2,906
Internal Stairs (1)	341	245	3,597	2,162
Internal Stairs (2)	337	245	3,601	2,146
Building Entrance	555	583	3,596	3,220
Main Drive	217	149	3,596	1,845
Main Drive (2)	184	145	3,597	1,760
Main Drive (3)	167	170	3,597	1,884
Main Drive (4)	109	142	3,597	1,469
Main Drive (5)	141	216	3,596	1,774
Main Drive (6)	169	157	3,596	1,872
Main Drive (7)	272	359	3,596	2,536
Level 3 External Lift	41	33	3,596	461
Level 4 Internal Lift	162	175	3,597	1,699
Level 4 External Lift	147	118	3,600	1,429
Total	5,854	5,690	79,133	42,109

TABLE I

PROPOSED DATABASE DETAILS: THE NUMBER OF PEDESTRIANS TRAVELLING IN EACH DIRECTION ARE GIVEN, ALONGSIDE THE TOTAL NUMBER OF SAMPLES EXTRACTED FROM EACH SEQUENCE WITH AND WITHOUT THE REMOVAL OF EXCESS SAMPLES THAT CONTAIN 0 PEOPLE. IT CAN BE SEEN THAT THE NUMBER OF PEOPLE PRESENT WITHIN THE CAMERAS VARIES GREATLY.

the secondary output in Section V-C; and finally compare to the scene dependent approach of [11] in Section V-D, and the scene independent approach of [15] in Section V-E.

A. Proposed Approach

Tables II and III present the performance of the proposed system across the 22 camera views. From Table II it is evident that the 2D CNN approach outperforms the simple LSTM network; and optical flow images offer better performance than grey scale images, while the use of both results in better performance than either one on their own. This is to be expected, as the additional size and complexity of the 2D CNN network over the LSTM, and the additional information offered by using both streams of data simultaneously, could reasonably be expected to offer improved performance. The benefit of the added complexity is particularly evident when considering the performance using grey scale images, where the 1D LSTM performs particularly poorly. The nature of the optical flow input, with the flow field being aligned such that only motion in the direction of interest (or it's opposite) is retained making people moving across the line appear more obvious, means that this input is able to work significantly better with a simpler network. However, we observe that the 1D LSTM is still able to effectively incorporate both modes of data to improve performance. Comparing these approaches

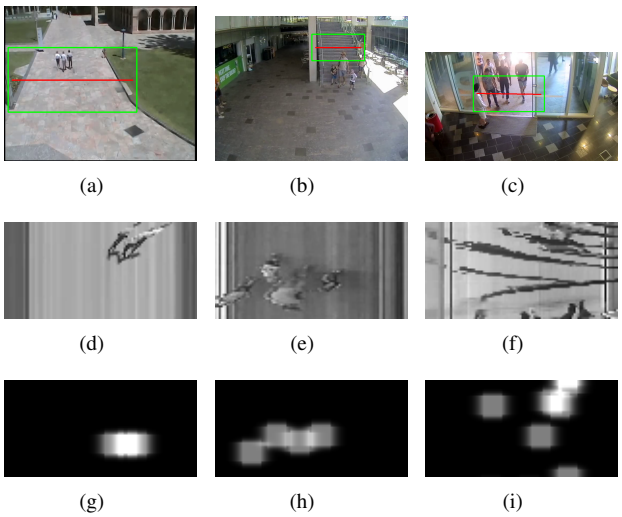


Fig. 8. Example location masks generated from the ground truth: The top row shows an input frame from within the relevant window; the second row shows the striped gray-scale image generated for the window; and the third row shows an example mask (forward direction in (g), reverse direction in (h) and (i)) for the window.

to the 2D CNN-LSTM network, we can see that the 2D CNN-LSTM achieves an average relative error (RE) close to that of the 2D CNN, although it's mean absolute error (MAE) and mean squared error (MSE) are worse than both the 2D CNN and 1D LSTM.

System	Data	MSE	MAE	RE
2D CNN	Optical Flow	0.34	0.24	52.16%
	Grey scale	0.60	0.39	78.54%
	Both	**0.32**	**0.23**	**42.43%**
1D LSTM	Optical Flow	0.46	0.27	56.12%
	Grey scale	1.04	0.71	354.89%
	Both	0.46	0.28	55.20%
2D CNN-LSTM	Both	0.56	0.33	45.71%

TABLE II

AVERAGE PERFORMANCE OF THE PROPOSED APPROACHES OVER THE 22 CAMERA CORPUS. NOTE THAT THE 2D CNN-LSTM APPROACH IS ONLY EVALUATED FOR BOTH SIMULTANEOUS OUTPUTS AS THIS HAS SHOWN BETTER PERFORMANCE FOR THE SIMPLER NETWORKS.

Table III shows the results for each sequence, and offers further insight into the performance of the systems. It can be seen that while the 2D CNN system achieves better overall error rates, there are a number of sequences where the 1D LSTM (External Lift, Building Entrance, Main Drive (4), and Level 4 Internal Lift) or the 2D CNN-LSTM (Pool Stairs, Level 3 Pathway, External Stairs (Bottom), Level 4 Pathway, Internal Stairs (1), Main Drive (3), Main Drive (5), Main Drive (6), Main Drive (7), Level 3 External Lift, Level 4 External Lift) achieve a more accurate overall estimate. From Figure 6, it can be seen that the 2D CNN typically offers best performance on those views that show a front-on view of the line of interest. Cameras that observe pedestrians from side-on, or partially side-on (i.e. Pool Stairs, Level 3 Pathway, Main Drive (3), Main Drive (4) and Main Drive (6)), or have significant perspective distortion (i.e. Building Entrance, Level 4 Internal Lift, Level 4 External Lift), tend to perform better with one (or both) of the LSTM networks.

Test Set	2D CNN			1D LSTM			2D CNN with LSTM		
	MSE	MAE	RE	MSE	MAE	RE	MSE	MAE	RE
Pool Stairs	**0.03**	**0.05**	26.68%	0.05	0.06	28.03%	0.04	0.05	**2.88%**%
Level 3 Pathway	0.08	**0.09**	35.40%	**0.07**	0.11	47.59%	0.09	0.12	**4.77%**%
External Stairs (Bottom)	**0.18**	**0.12**	39.28%	0.26	0.14	42.50%	0.24	0.15	**0.15%**%
Beside External Stairs (Bottom)	**0.09**	**0.14**	**10.72%**	0.35	0.27	65.58%	0.14	0.19	16.12%%
External Lift	0.06	**0.06**	122.92%	**0.04**	0.07	**110.98%**	0.08	0.13	310.04%%
Level 4 Pathway	**0.29**	**0.20**	52.94%	0.40	0.22	58.00%	0.58	0.34	**25.61%**%
Main Door (1)	**1.23**	**0.72**	**18.83%**	1.90	0.80	46.45%	2.93	1.07	57.32%%
Main Door (2)	**1.20**	**0.69**	**19.11%**	1.49	0.73	30.46%	2.24	0.86	58.14%%
External Stairs (Top)	**0.26**	**0.26**	**8.51%**	0.62	0.39	60.86%	0.46	0.35	23.51%%
Internal Stairs (1)	**0.36**	**0.24**	39.64%	0.61	0.31	72.13%	0.48	0.30	**32.16%**%
Internal Stairs (2)	**0.28**	0.26	**6.89%**	0.29	**0.21**	25.56%	0.52	0.28	42.06%%
Building Entrance	**0.40**	**0.30**	30.22%	0.51	0.38	**9.18%**	1.45	0.72	57.69%%
Main Drive	**0.22**	**0.20**	**1.63%**	0.42	0.24	82.20%	0.35	0.28	9.63%%
Main Drive (2)	**0.14**	**0.12**	**24.34%**	0.33	0.27	27.94%	0.21	0.20	24.91%%
Main Drive (3)	0.19	0.15	62.12%	**0.15**	**0.14**	35.93%	0.18	0.17	**17.08%**%
Main Drive (4)	**0.12**	**0.10**	49.99%	0.13	0.17	**9.20%**	0.28	0.31	119.12%%
Main Drive (5)	**0.15**	**0.14**	32.46%	0.35	0.22	79.61%	0.29	0.22	**30.43%**%
Main Drive (6)	**0.24**	**0.18**	70.40%	0.29	0.21	72.35%	0.34	0.28	**17.47%**%
Main Drive (7)	0.81	0.42	88.75%	0.92	0.43	98.24%	**0.56**	**0.42**	**27.17%**%
Level 3 External Lift	**0.05**	**0.06**	77.33%	0.05	0.08	108.13%	0.06	0.06	**25.77%**%
Level 4 Internal Lift	0.30	0.31	75.56%	**0.30**	**0.28**	**40.32%**	0.50	0.51	101.99%%
Level 4 External Lift	0.39	0.28	39.76%	0.49	0.33	90.17%	**0.33**	**0.27**	**1.59%**%
Average	**0.32**	**0.23**	**42.43%**	0.46	0.28	56.43%	0.56	0.33	45.71%

TABLE III

PERFORMANCE OF THE PROPOSED APPROACH FOR EACH SEQUENCE IN THE DATABASE. WHILE THE 2D CNN WORKS BETTER OVERALL, FOR A NUMBER OF SEQUENCES THE SIMPLER LSTM OFFERS BETTER PERFORMANCE.

From these views, occlusions as multiple people cross the gate are more common, and thus strong visual features that indicate a single person is crossing the gate, such as those that may be observed in front-on views, are less prevalent. For the 2D CNN approach which relies on being able to learn a set of filters that correspond to an individuals appearance as they cross the gate, this means that counts are less accurate. This is further illustrated in Figure 9, where we can see that for two front-on gates (Level 4 Pathway, Main Door (2)) the LSTMs are more prone to under-counting; while from the partially or fully side-on views (Building Entrance, Main Drive (3)), the 2D CNN suffers more greatly from under-counting. From Figure 9 we can also see that the performance of the 2D CNN-LSTM is quite inconsistent across the views. Generally, it struggles to count large groups of people (Main Drive (3), on which it performs best of the five sequences we visualise has the smallest size crowds), and while it achieves the lowest relative error on 'Level 4 Pathway' of the three systems, we can see that this is actually due to a small positive bias that accumulates to effectively help make up for the under-counting towards the middle of the sequence. We observe in general that all networks struggle somewhat with under-counting large groups, which can be attributed to the distribution of crowd sizes in the database (see Figure 7) where only a very small number of training examples with 10 or more people present exist. However, the 2D CNN-LSTM seems even more sensitive to this problem than the other networks, suggesting that it is struggling to learn an effective set of filters to detect a person as they cross the gate.

This problem for the 2D CNN-LSTM may be caused by the wide variation in size of the gates in real world terms, as shown in Figure 6. The cropped region extracted for some gates (such as Main Door (1), Main Door (2), Building Entrance) will not actually contain an entire person at once, which likely hampers the 2D CNN-LSTM approach in it's learning. This is further supported by the performance on 'Main Drive (5)' and 'Main Drive (7)', where the 2D CNN-LSTM performs best, despite these being front-on gates that typically work best with the 2D CNN. Compared to other front-on views, these are both very wide gates in real-world terms, from Figure 6 we can see that entire people are present in the extracted regions, helping the network detect and count people in the sequences.

All three networks are capable of estimating crowd sizes quickly, with inference times per sample of 0.0433 seconds, 0.0120 seconds, and 0.2913 seconds for the 2D CNN, 1D LSTM and 2D CNN-LSTM networks respectively, running on a single core of a Xeon 2670 CPU. Unsurprising, the 1D LSTM is the quickest, while the 2D CNN-LSTM is the slowest; however given that a single sample represents a 10 second window, all three are performing well above real-time on only a single CPU.

B. Loss Function Evaluation

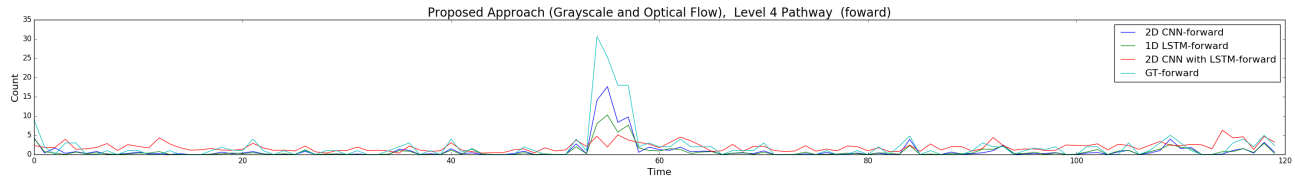
An evaluation of three different loss functions for the primary output (i.e. the counts) is shown in Table IV. It can be seen that using the combination of mean squared error (MSE) and mean absolute error (MAE) clearly offers superior performance.

Loss Function	MSE	MAE	RE
MSE + MAE	**0.32**	**0.23**	**42.43%**
MSE	0.36	0.33	98.03%
MAE	0.36	0.23	46.10%

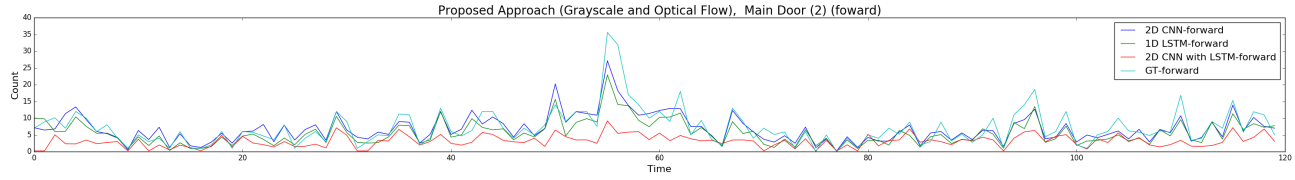
TABLE IV

AVERAGE PERFORMANCE ACROSS 22 FOLDS OF DIFFERENT LOSS FUNCTIONS FOR THE 2D CNN. NETWORKS USE BOTH GREY SCALE AND OPTICAL FLOW IMAGES AS INPUT.

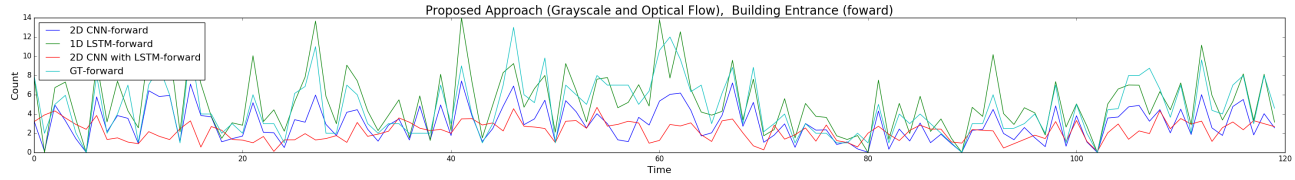
Figure 10 illustrates the reason for this. Using the MSE as a loss function will result in small counting errors (i.e.



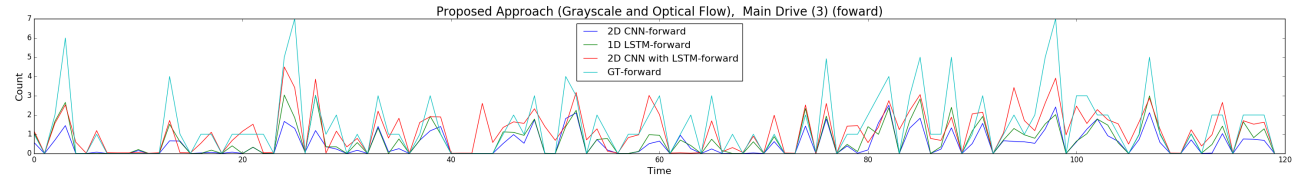
(a) Level 4 Pathway, Forward Direction



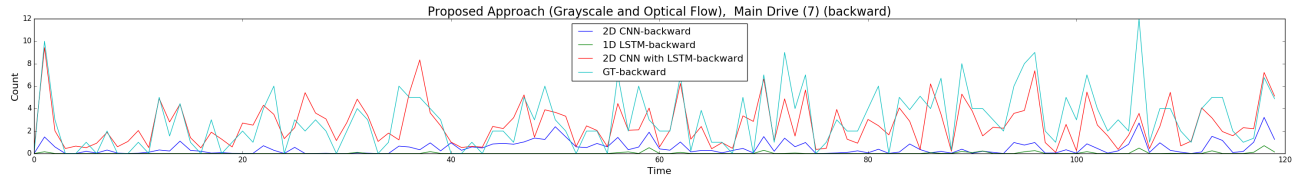
(b) Main Door (2), Forward Direction



(c) Building Entrance, Forward Direction



(d) Main Drive (3), Forward Direction



(e) Main Drive (7), Backward Direction

Fig. 9. Performance of the proposed approaches for a selection of sequences. Performance varies for the different networks across the different views.

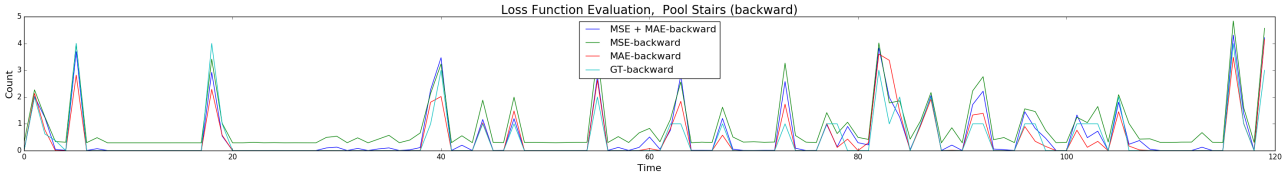
less than 1.0) being suppressed by the loss function; with far greater emphasis given to errors when estimating windows that contain a large number of people. As such the system is prone to learning a small positive bias. Using the MAE resolves this problem and the system is largely free of bias and correctly detects when no one is present, however the networks are now more prone to under-counting. Using the sum of the two metrics as the loss somewhat alleviates the under-counting, while retaining the ability to correctly detect windows without a person (i.e. little to no bias). We note that all metrics are somewhat prone to under-counting on some sequences such as 'Main Door (2)' where large crowds are present. As noted earlier, this can be attributed to the nature of the dataset where there are very few examples of windows with 10 or more people.

C. Impact of Secondary Output

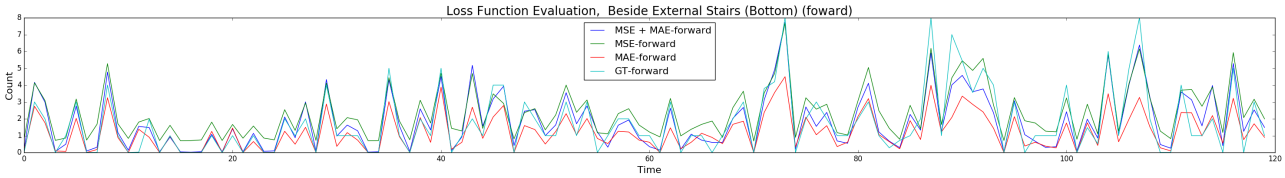
Table V and Figure 11 show the impact of the secondary output on system performance. From Table V, it is clear that the secondary output has a beneficial impact on performance. Looking at Figure 11, we can see that the secondary output has two main benefits: assisting with the correct identification of situations where 0 people are present, as evidenced by the small bias present in the single output networks shown in Figure 11 (a) and (c); and improving the detection and counting of larger groups, in particular for the LSTM network (see Figure 11 (b)).

System	MSE	MAE	RE
2D DCNN (1 Output)	0.32	0.25	52.93%
1D LSTM (1 Output)	0.47	0.29	81.51%
2D DCNN (2 Outputs)	**0.32**	**0.23**	**42.43%**
1D LSTM (2 Outputs)	0.46	0.28	55.20%

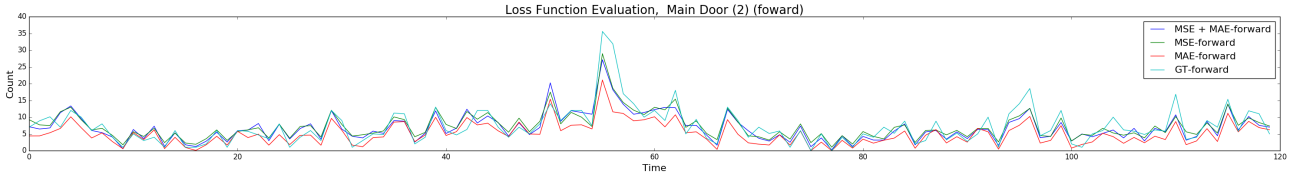
TABLE V
AVERAGE PERFORMANCE ACROSS 22 FOLDS FOR NETWORK THAT DO OR DON'T USE THE SECONDARY OUTPUT. ALL NETWORKS USE BOTH GREY SCALE AND OPTICAL FLOW IMAGES.



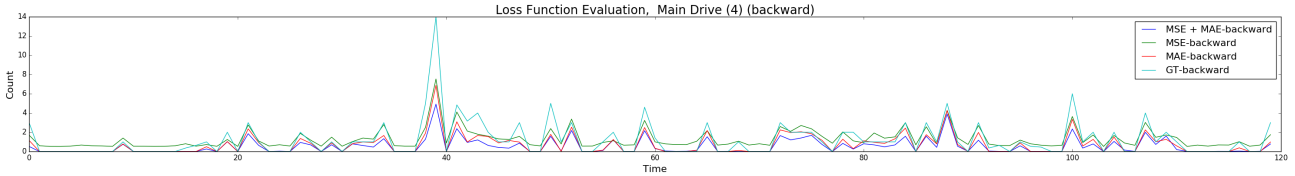
(a) Pool Stairs, Backward Direction



(b) Beside External Stairs (Bottom), Forward Direction

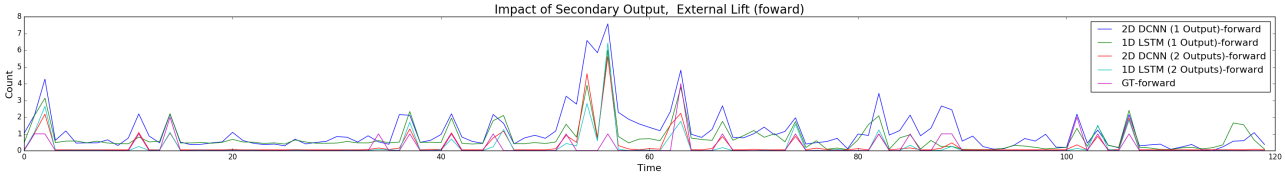


(c) Main Door (2), Forward Direction

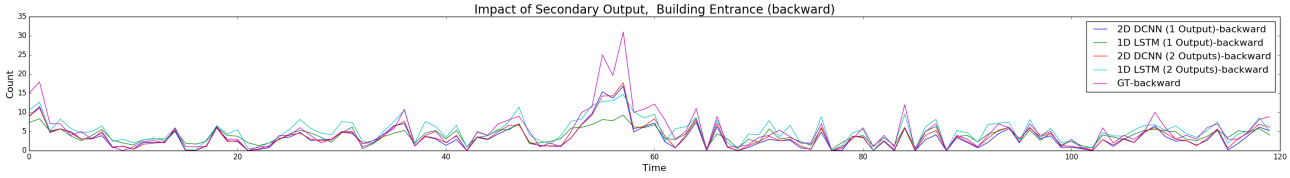


(d) Main Drive (4), Backward Direction

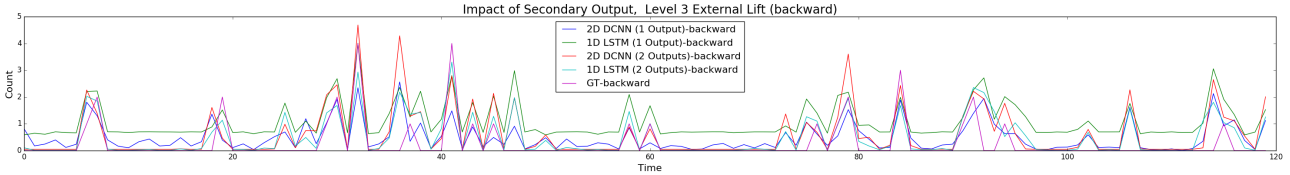
Fig. 10. Performance of the 2D CNN with different loss functions for different sequences. It can be seen that using MSE alone leaves the system more prone to learning a small positive bias. Use of the MAE alone results in the correct counting of observations that contain 0 people, but at the cost of more severe under-counting.



(a) External Lift, Forward Direction



(b) Building Entrance, Forward Direction



(c) Level 3 External Lift

Fig. 11. Performance of the proposed networks (2D CNN and LSTM) with and without the secondary output. All networks use both grey scale and optical flow images.

D. Comparison to Scene Dependant Approach

Table V-D compares the proposed approach (2D CNN network) to the scene dependant technique of [11]. As the approach of [11] is scene dependant, the first half of each sequence (i.e. one hour) is used to train the system, and it is tested on the second hour. The proposed approach is trained on the data from the other sequences, as per the rest of the evaluation. Results are reported for all systems on the second hour of footage only, and as such results for the proposed system are different to those results reported in Table III.

From Table V-D, we can see that the proposed approach typically achieves lower MSE and MAE than the scene dependant technique, however when assessing based on the relative error, performance is somewhat more mixed. The baseline scene dependant technique performs badly on views with lots of clutter and occlusions, such as 'Pool Stairs', 'Level 3 Pathway', 'External Lift' and 'Building Entrance'. Plots for the first two of these sequences are shown in Figure 12 (a) and (b), and it can be seen that in the presence of such clutter and spurious motion, large false counts are recorded. This is particularly problematic in 'External Lift', where the gate is frequently obscured by large crowds of pedestrians moving across the pathway in a direction perpendicular to that of the gate.

Performance is more mixed for the more front-on and less obscured gates. Looking at 'Main Door (1)' and 'Main Door (2)' shown in Figure 12 (c) and (d), we can see that the proposed approach performs better on 'Main Door (1)', but not 'Main Door (2)'. This is due to the crowded nature of these gates, and as noted earlier, the lack of data with similar crowd levels in the rest of the dataset. The scene dependant approach, having been trained on the previous hour of footage and thus similar crowd densities, is better able to cope with the crowded conditions of 'Main Door (2)'. In Figure 12 (c) it can be seen that although the larger crowds at the end of the sequence are under-counted by the proposed approach, the more accurate estimation throughout the remainder of the sequence leads to a more accurate performance overall.

The tendency of the proposed approach to undercount in many situations hampers it's performance in sequences such as 'Main Drive (5)', as seen in Figure 12 (e). The proposed and scene dependent approach both follow the trends present in the ground truth, however the proposed approach typically reports fewer people than the baseline approach, leading to a gradual accumulation of error. In spite of this, we note that the proposed approach achieves lower MSE and MRE than the baseline for this sequence, illustrating how these metrics, when taken on their own, can be misleading.

Finally, we consider the performance of the techniques on 'Level 4 Internal Lift', as shown in Figure 12 (f). This view is significantly different from other views in the dataset, with the presence of perspective distortion, and the comparatively unusual behaviour of people stopping and loitering around the line of interest as they wait for the lift. In a situation significantly different from many others observed in the database, the proposed approach struggles, while the baseline technique obtains a more accurate overall count.

Test Set	View Specific			Proposed		
	MSE	MAE	RE	MSE	MAE	RE
Pool Stairs	0.12	0.17	73.63%	**0.03**	**0.05**	**27.83%**
Lev. 3 Path.	0.56	0.56	346.52%	**0.08**	**0.09**	**32.94%**
Ext. Stairs (B)	**0.17**	0.22	**8.86%**	0.19	**0.12**	38.89%
Bsd. Ext. Stairs (B)	0.23	0.26	**2.93%**	**0.09**	**0.15**	9.71%
Ext. Lift	0.20	0.35	1104.38%	**0.06**	**0.06**	**134.35%**
Lev. 4 Path.	**0.29**	0.23	69.41%	0.30	**0.20**	**52.96%**
Main Door (1)	**1.17**	0.73	21.10%	1.26	**0.72**	**18.36%**
Main Door (2)	**0.83**	**0.63**	**0.39%**	1.22	0.69	19.30%
Ext. Stairs (T)	0.37	0.39	12.16%	**0.27**	**0.26**	**10.12%**
Int. Stairs (1)	0.69	0.41	**3.28%**	**0.36**	**0.23**	39.46%
Int. Stairs (2)	**0.20**	**0.23**	7.41%	0.26	0.26	**5.75%**
Building Entrance	5.17	1.85	231.77%	**0.40**	**0.30**	**30.23%**
Main Drv	0.29	0.35	38.60%	**0.24**	**0.22**	**2.07%**
Main Drv (2)	0.21	0.26	69.21%	**0.15**	**0.13**	**24.26%**
Main Drv (3)	2.19	0.43	76.81%	**0.20**	**0.16**	**61.76%**
Main Drv (4)	0.34	0.29	**38.87%**	**0.13**	**0.10**	49.50%
Main Drv (5)	**0.16**	0.20	**5.25%**	0.16	**0.14**	31.26%
Main Drv (6)	0.36	0.27	**70.20%**	**0.25**	**0.19**	70.80%
Main Drv (7)	**0.61**	0.52	**8.78%**	0.86	**0.44**	88.68%
Lev. 3 Ext. Lift	0.06	0.07	**26.35%**	**0.05**	**0.06**	78.12%
Lev. 4 Int. Lift	0.35	**0.26**	**5.83%**	**0.29**	0.30	73.24%
Lev. 4 Ext. Lift	**0.32**	0.31	**28.54%**	0.40	**0.29**	42.18%
Average	0.68	0.41	102.29%	**0.33**	**0.23**	**42.81%**

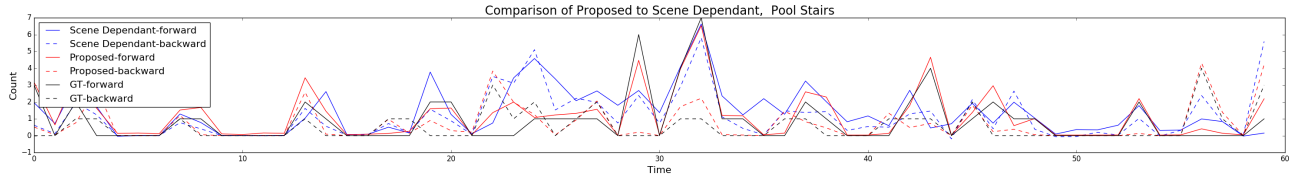
TABLE VI
COMPARISON OF THE PROPOSED APPROACH TO THE SCENE DEPENDANT APPROACH OF [11].

E. Comparison to Scene Independent Approaches

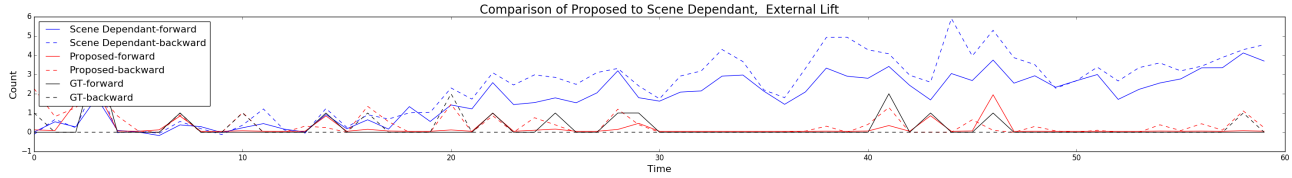
We compare the proposed approach to the scene invariant, deep learning method of Cao et al. [15][3], and the scene dependant approach of [11] trained in a scene invariant method. Both systems are trained in the same manner as the proposed approach, using 17 of the 22 sequences as the training set. The 4 validation sequences are not used by the approach of [11]. For the approach is Cao et al. [15], data is rescaled to the target size specified in their paper, each of the three networks are trained independently and the networks are selected using 4 validation sequences. Results for the systems are shown in Table VII and Figure 13.

From these results, it can be seen that the proposed approach outperforms the baselines on the majority of sequences, particularly in terms of MAE and MSE. The baselines achieve somewhat mixed results, with both performing very poorly on a number of sequences. This is to be expected for [11]

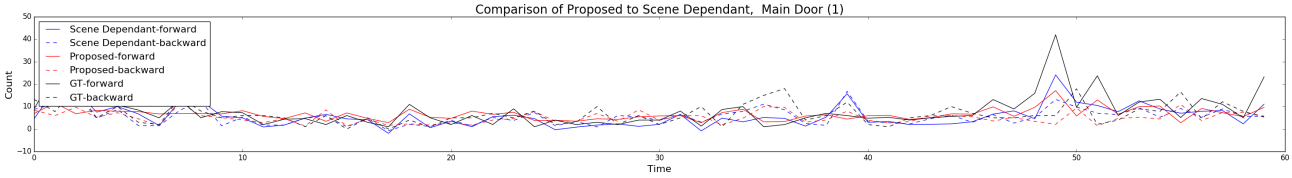
[3]We re-implement [15] following the architecture described in their paper



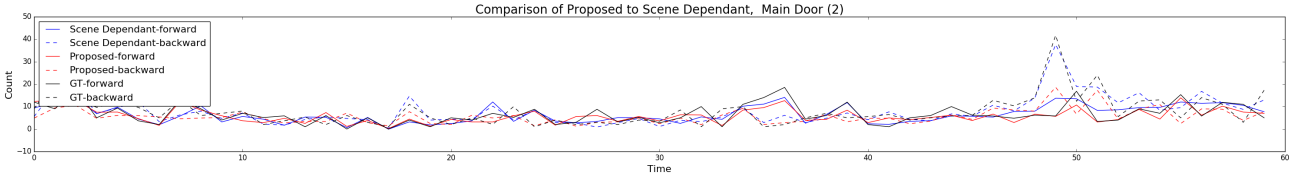
(a) Pool Stairs



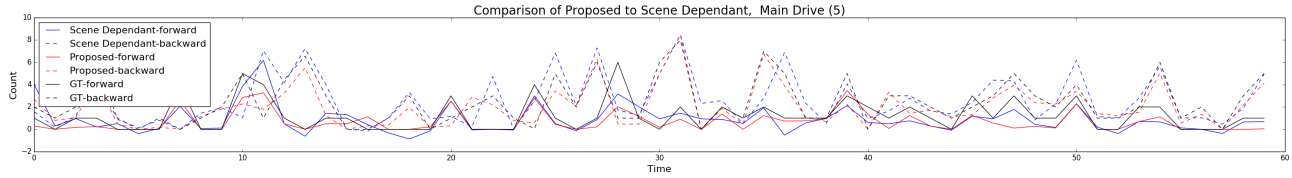
(b) External Lift



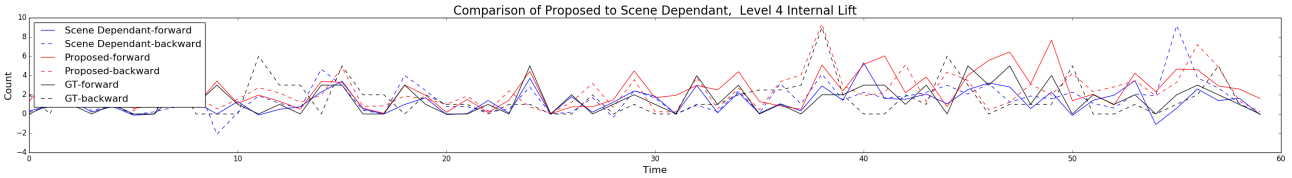
(c) Main Door (1)



(d) Main Door (2)



(e) Main Drive (5)



(f) Level 4 Internal Lift

Fig. 12. Performance of the proposed approach compared to the scene dependent technique of [11] for a selection of scenes.

as it is a scene dependant technique, however we notice that on test views where a similar view is present in the training set (i.e. the pairs 'External Stairs (Top)' and 'External Stairs (Bottom)'; and 'Main Drive (3)' and 'Main Drive (6)') performance is good and can outperform the proposed approach in terms of relative error. From Figure 13, it can be seen that in some cases such as for 'Main Drive (3)' (see Figure 13, (h)) this is due to a learned bias that fortuitously leads to an approximately correct estimate over the sequence, while for others (i.e. 'External Stairs (Top)') it is able to learn a somewhat scene invariant representation, though some bias in the forward direction does still appear to be present (see Figure 13, (c)).

The deep learning approach of Cao et al. [15] also performs well for a number of sequences, and typically outperforms [11] in terms of MAE and MSE. We note that Cao et al. [15] is very effective at detecting when the gate is empty and avoiding spurious counts; however the cost of this is that on the whole this approach tends to undercount, and does at times learns a representation that fails to count any people at all (rows in Table VII with a 100% RE). The nature of the approach, where three networks are used to estimate the total crowd count, mode of crowd motion and the ratio of people entering to leaving is likely the source of this limitation; as errors

Test Set	Denman et al. [11]			Cao et al. [15]			Proposed		
	MSE	MAE	RE	MSE	MAE	RE	MSE	MAE	RE
Pool Stairs	0.25	0.39	335.87%	0.09	0.07	55.29%	**0.03**	**0.05**	**26.68%**
Level 3 Pathway	0.18	0.28	193.08%	0.44	0.31	176.50%	**0.08**	**0.09**	**35.40%**
External Stairs (Bottom)	0.23	0.16	**23.23%**	0.37	0.19	44.34%	**0.18**	**0.12**	39.28%
Beside External Stairs (Bottom)	0.25	0.33	35.04%	0.72	0.39	11.48%	**0.09**	**0.14**	**10.72%**
External Lift	0.09	0.22	414.42%	**0.05**	0.06	**35.25%**	0.06	**0.06**	122.92%
Level 4 Pathway	0.51	0.44	**19.83%**	0.78	0.30	100.00%	**0.29**	**0.20**	52.94%
Main Door (1)	2.01	1.08	54.69%	3.33	1.16	48.12%	**1.23**	**0.72**	**18.83%**
Main Door (2)	1.83	0.91	25.14%	2.28	1.00	36.26%	**1.20**	**0.69**	**19.11%**
External Stairs (Top)	0.64	0.55	**3.55%**	1.32	0.56	100.00%	**0.26**	**0.26**	8.51%
Internal Stairs (1)	0.58	0.37	52.03%	0.65	0.35	39.64%	**0.36**	**0.24**	**39.64%**
Internal Stairs (2)	0.64	0.56	95.99%	0.39	**0.25**	23.46%	**0.28**	0.26	**6.89%**
Building Entrance	0.87	0.65	43.29%	0.50	0.36	**16.06%**	**0.40**	**0.30**	30.22%
Main Drive	2.03	1.05	394.76%	0.35	0.24	48.79%	**0.22**	**0.20**	**1.63%**
Main Drive (2)	0.29	0.39	111.36%	0.28	0.15	50.78%	**0.14**	**0.12**	**24.34%**
Main Drive (3)	1.45	0.44	**6.01%**	0.23	0.17	68.61%	**0.19**	**0.15**	62.12%
Main Drive (4)	0.35	0.53	229.46%	0.17	0.11	61.01%	**0.12**	**0.10**	**49.99%**
Main Drive (5)	0.90	0.53	32.95%	0.52	0.25	100.00%	**0.15**	**0.14**	**32.46%**
Main Drive (6)	0.35	0.30	**45.31%**	0.42	0.22	100.00%	**0.24**	**0.18**	70.40%
Main Drive (7)	1.12	0.80	32.50%	0.93	0.50	**27.63%**	**0.81**	**0.42**	88.75%
Level 3 External Lift	0.72	0.64	606.40%	0.11	0.06	**42.05%**	**0.05**	**0.06**	77.33%
Level 4 Internal Lift	0.47	0.40	63.46%	0.43	**0.27**	**21.43%**	**0.30**	0.31	75.56%
Level 4 External Lift	0.79	0.80	387.60%	0.46	0.30	**23.78%**	**0.39**	**0.28**	39.76%
Average	0.75	0.54	145.73%	0.67	0.33	55.93%	**0.32**	**0.23**	**42.43%**

TABLE VII
PERFORMANCE OF THE PROPOSED APPROACH COMPARE TO THE SCENE INVARIANT APPROACH OF CAO ET AT. [15] AND THE APPROACH OF DENMAN ET AL. [11] TRAINED IN A SCENE INVARIANT MANNER.

in any one of the networks can lead to counting errors. The proposed approach, which uses both inputs jointly within the one network, effectively reduces the number of ways in which counting errors can occur, thus leading to increased accuracy. We do note however that the approach of Cao et al. [15] generally performs much better in the presence of perspective distortion. The sequences 'External Lift', 'Building Entrance', 'Level 3 External Lift', 'Level 4 Internal Lift' and 'Level 4 External Lift' all contain significant perspective distortion, and Cao et al's [15] approach achieves impressive results on all of these.

Overall, we see the proposed approach performs more consistently than the other two techniques. While there are some scenes where it is outperformed by the other techniques, we don't observe any of the complete failure that at times occurs with the other systems, such as the failure to count anyone, or the presence of a bias leading to large over-counting.

VI. CONCLUSION

In this paper we have proposed a scene invariant approach for pedestrian throughput estimation using deep networks and LSTMs. The proposed approach has been demonstrated on a new 44 hour database, that captures over $11,000$ pedestrian movements from 22 distinct views cover indoor and outdoor scenes and a variety of camera angles and fields of view. Using this database, we demonstrate the efficacy of the proposed approach, achieving similar or better performance to a scene dependent approach in a majority of views - particularly those which contain significant clutter and occlusions and are otherwise difficult to count. We also show improved performance over an existing deep-learning based pedestrian throughput technique [15] on the proposed database.

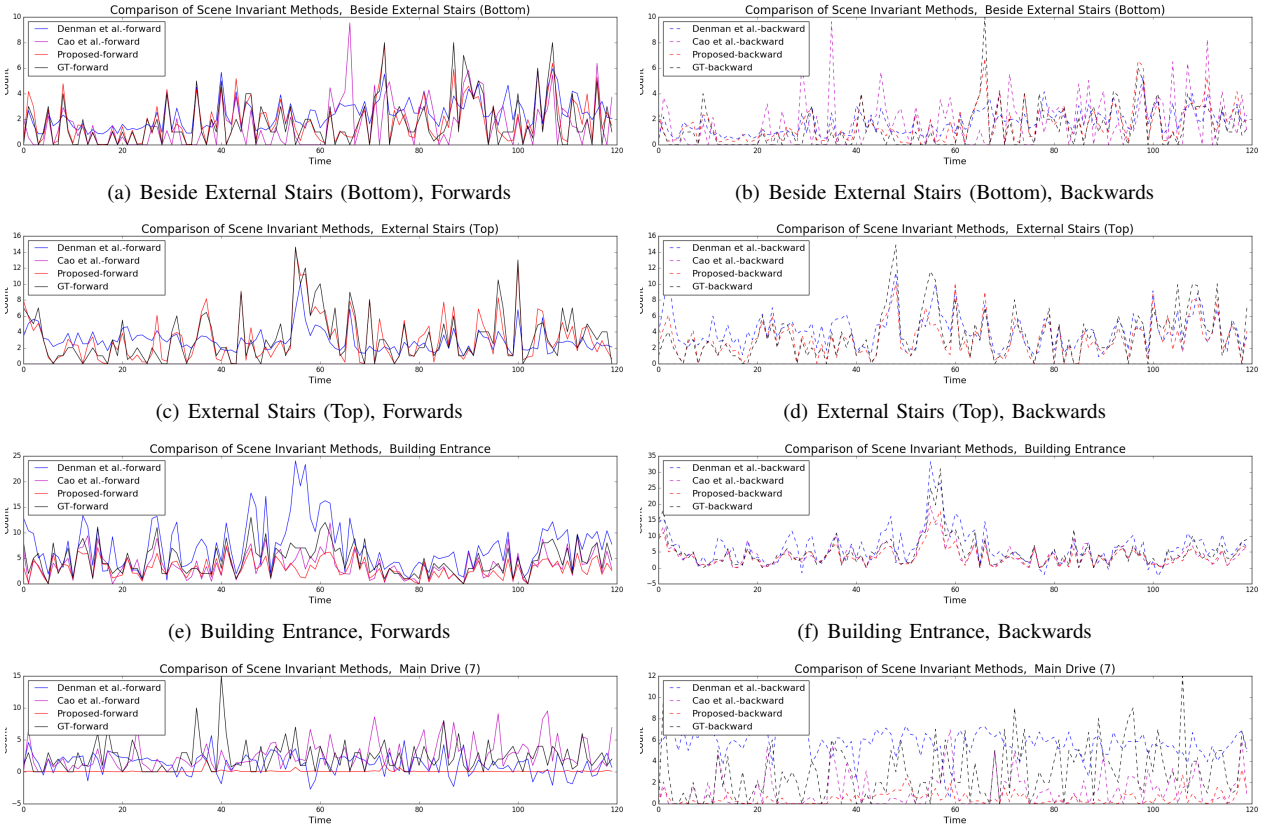
We find that while the inclusion of LSTMs within the network does not lead to an overall gain in performance, we do nonetheless see improvements for full or partially side-on gates and those with perspective distortion, suggesting further investigation is warranted. Future work will focus on how both the simplistic 2D representation and the spatio-temporal volume representation can be jointly used with simple supplementary information about the scene such as coarse gate pose information (i.e. front on, side-on, presence of perspective or not) to leverage the relative strengths of both methods and improve overall performance. Other approaches such as 3D convolutional neural networks will also be explored to better extract temporal information, and additional data will continue to be collected to further expand the database, and evaluate the proposed and future techniques. Recent developments in crowd counting with deep neural networks will also be investigated, such as the use of deep metric learning [24] and the multi-scale regression approach of [25], as potential methods to further improve performance. Finally, comparisons to other approaches such as [16] will be made as data becomes available.

ACKNOWLEDGEMENT

This research was supported by the Australian Research Council's Linkage Project "Improving Productivity and Efficiency of Australian Airports" (140100282). The authors would also like to thank QUT High Performance Computing (HPC) for providing the computational resources for this research.

REFERENCES

[1] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "A method of counting the passing people by using the stereo images," in *ICIP*, vol. 2, 1999, pp. 338–342 vol.2.



(a) Beside External Stairs (Bottom), Forwards

(b) Beside External Stairs (Bottom), Backwards

(c) External Stairs (Top), Forwards

(d) External Stairs (Top), Backwards

(e) Building Entrance, Forwards

(f) Building Entrance, Backwards

(g) Main Drive (3), Forwards

(h) Main Drive (3), Backwards

Fig. 13. Performance of the proposed approach compared to the techniques of [11] and [15] for a selection of scenes.

[2] J.-W. Kim, K.-S. Choi, B.-D. Choi, and S.-J. Ko, "Real-time vision-based people counting system for the security door," in *International Technical Conference on Circuits/Systems Computers and Communications*, 2002, pp. 1416–1419.

[3] T.-H. Chen, "An automatic bi-directional passing-people counting method based on color image processing," oct. 2003, pp. 200 – 207.

[4] T.-H. Chen, T.-Y. Chen, and Z.-X. Chen, "An intelligent people-flow counting method for passing through a gate," in *IEEE Conference on Robotics, Automation and Mechatronics*, June 2006, pp. 1–6.

[5] S. Velipasalar, Y.-L. Tian, and A. Hampapur, "Automatic counting of interacting people by using a single uncalibrated camera," in *International Conference on Multimedia and Expo*, July 2006, pp. 1265–1268.

[6] J. Barandiaran, B. Murguia, and F. Boto, "Real-time people counting using multiple lines," may 2008, pp. 159 –162.

[7] A. Albiol, A. Albiol, and J. Silla, "Statistical video analysis for crowds counting," in *ICIP*, 2009, pp. 2569–2572.

[8] B.-S. Kim, G.-G. Lee, J.-Y. Yoon, J.-J. Kim, and W.-Y. Kim, "A method of counting pedestrians in crowded scenes," in *ICIC*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 1117–1126.

[9] Z. Ma and A. B. Chan, "Crossing the line: Crowd counting by integer programming with local features," in *CVPR*, 2013, pp. 2539–2546.

[10] S. Mukherjee, S. Gil, and N. Ray, "Unique people count from monocular videos," *The Visual Computer*, vol. 31, no. 10, pp. 1405–1417, 2015.

[11] S. Denman, C. Fookes, D. Ryan, and S. Sridharan, "Large scale monitoring of crowds and building utilisation: A new database and distributed approach," in *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[12] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Scene invariant multi camera crowd counting," *PRL*, vol. 44, pp. 98–112, 2014.

[13] B. Liu and N. Vasconcelos, "Bayesian model adaptation for crowd counts," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[14] K. Kang and X. Wang, "Fully convolutional neural networks for crowd segmentation," *arXiv preprint arXiv:1411.4464*, 2014.

[15] L. Cao, X. Zhang, W. Ren, and K. Huang, "Large scale crowd analysis based on convolutional neural network," *Pattern Recognition*, vol. 48, no. 10, pp. 3016–3024, 2015.

[16] Z. Zhao, H. Li, R. Zhao, and X. Wang, *Crossing-Line Crowd Counting with Two-Phase Deep Neural Networks*. Cham: Springer International Publishing, 2016, pp. 712–726.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[21] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," *CoRR*, vol. abs/1411.4280, 2014. [Online]. Available: http://arxiv.org/abs/1411.4280

[22] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015. [Online]. Available: http://arxiv.org/abs/1511.07289

[23] T. Dozat, "Incorporating nesterov momentum into adam," Stanford University, Tech. Rep., 2015.[Online]. Available: http://cs229. stanford. edu/proj2015/054 report. pdf, Tech. Rep., 2015.

[24] Q. Wang, J. Wan, and Y. Yuan, "Deep metric learning for crowdedness regression," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[25] D. Oñoro-Rubio and R. J. López-Sastre, *Towards Perspective-Free Object Counting with Deep Learning*. Cham: Springer International Publishing, 2016, pp. 615–629.