Universidad de Concepción
Graduate School
Faculty of Engineering - Electrical Engineering Master's Program

Dual Reconstructive Autoencoder for Crowd Localization and Estimation in Density and FIDT Maps

Thesis to qualify for the degree of
Master of Science in Electrical Engineering

FELIPE IGNACIO LAMAS SILVA
CONCEPCIÓN - CHILE
OCTOBER 2022

Advisor: Sebastián Godoy Medel
Electrical Engineering Department, Faculty of Engineering
Universidad de Concepción

To the three most

influential women in my life:

My mom, my grandma, and my fiancée;

and in memory of my dear friend

Professor Jorge E. Pezoa

Abstract

This research proposes a novel crowd estimation technology to help authorities to make the right decisions in times of crisis. Specifically, deep learning models have faced these challenges, achieving excellent results. In particular, the trend of using single-column Fully Convolutional Networks (FCNs) has increased in recent years. A typical architecture that meets these characteristics is the autoencoder. However, this model presents an intrinsic difficulty: the search for the optimal dimensionality of the latent space. In order to alleviate such difficulty, we propose a dual architecture consisting of two cascaded autoencoders. The first autoencoder is responsible for carrying out the masked reconstruction of the original images, whereas the second obtains crowd maps from the outputs of the first one. Our architecture improves the location of people and crowds on Focal Inverse Distance Transform (FIDT) maps, resulting in more accurate count estimates than estimates obtained through a single autoencoder architecture. Specifically, to evaluate the model in the location task we used two decision thresholds ($\sigma_1 = 4$ and $\sigma_2 = 8$), obtaining, respectively, that our model increased the Precision by 36 (from 27.11% to 63.11%) and 46.8 (from 37.26% to 84.06%) percentage points, the Recall metric by 3.05 (from 54.56% to 57.61%) and 1.75 (from 74.98% to 76.73%) percentage points, and F1-Score by 24.02 (from 36.22% to 60.24%) and 30.45 (from 49.78% to 80.23%) percentage points. For the counting task, the Dual Reconstructive Autoencoder (DRA) model decreased MAE and RMSE by 88.5% and 75.18%, respectively, compared to the metrics obtained for the Single Autoencoder (SA) model (SA model MAE: 121.73, DRA model MAE: 13.92, SA model RMSE: 127.61, DRA model RMSE: 31.67).

Contents

List of Figures

List of Tables

Acknowledgments

This work was supported by Chilean National Agency for Research and Development (ANID) / National Natural Science Foundation of China (NSFC) under the International Cooperation Program grant number PII180009, and ANID-Subdirección de Capital Humano/Magíster Nacional/2021 - 22210563 fellowship.

Thanks,

Felipe Lamas Silva

Chapter 1

Introduction

The global figures for Covid-19 infections show the rapid spread of the virus in Chile [1] and the world [2]. It is known that crowded spaces are directly related to high infection rates. At the beginning of the pandemic, health authorities used various mechanisms to avoid crowds, such as the permanent closure of shopping centers, curfews, preventive and mandatory quarantines, and teleworking. However, these mechanisms notably harmed global economies and the general welfare of the population. Given the successful vaccination campaigns, it has been possible to return to the routine, maintaining social distancing and using masks. However, there are still crowds of people who do not respect the protocols for different reasons, which are generally very difficult to handle. As such, crowd detection and management are still critical.

Similarly, crowd management is highly critical in natural disasters. Earthquakes, tsunamis, forest fires, floods, and mudslides are some natural phenomena that frequently cause stampedes of uncontrolled people. Two of the countries most exposed to natural catastrophes are China and Chile [3]. Specifically, earthquakes are frequent threats in both countries, generating fatalities and considerable material losses [4].

Automatic people counting technologies can help authorities to make vital decisions in difficult moments to reduce civilian casualties. Manual counting of people from a video feed of a security camera is not an option since, in general, people in these scenes change constantly. Moreover, manual counting is a time-consuming task, and usually, the count is required in almost real-time. As such, machine-learning approaches are required to tackle this problem.

1.1 Previous Work

There are three main machine-learning approaches in crowd estimation: detection, regression, and density maps [5]. The detection approach was the first to appear and was mainly characterized by sliding window detectors [6, 7, 8, 9]. This approach fails when many occluded people are in the image. This problem was solved using texture and foreground feature-extraction regression methods [10, 11].

On the other hand, due to the success of deep learning, specifically artificial neural net-

works, many contemporary authors use their convolutional versions to develop models capable of counting people through hierarchical learning of the data characteristics and then making high-quality inferences from them. A typical classification for convolutional neural networks is basic, single-column, and multi-column. The basic networks, denoted as Convolutional Neural Network (CNN), were the first to appear and have fully connected layers (dense layers) at the end of their architecture. On the other hand, single-column and multi-column networks do not have dense layers and estimate the number of people directly from the output density map of the network. These models are commonly called fully convolutional networks (FCN), and the difference between these two types lies in the number of columns used by the architecture.

C. Wang *et al.* [12] developed one of the first works on this subject, obtaining the count of people in highly dense images through a regression model that used a basic convolutional neural network. However, some researchers realized that density maps contribute much more to people counting than estimating them directly from images because the models can learn the characteristics of crowds through them [13]. Accordingly, Y. Zhang *et al.* [13] employed a multi-column convolutional neural network to obtain density maps to accurately estimate the number of people in images with arbitrary perspectives and crowd distributions. However, as was demonstrated by [14], the network is not very effective when using columns with different receptive fields since each one learns practically the same characteristics of the images.

Subsequently, L. Boominathan *et al.* [15] used a multi-column convolutional neural network to estimate density maps and total people in images of significantly dense crowds. Combining a deep and shallow network allows scale-invariant detection from widely occluded images. A similar approach to the ones used in [13, 15] is the one implemented by L. Zeng *et al.* [16], where they employed a single-column, high-performance multi-scale model based on the generation of scale-relevant features.

M. Marsden *et al.* [17] used a single column deep neural network to obtain density maps and achieve high accuracy in people counting. Researchers can examine images of any resolution and aspect ratio with such a model. Also, a 50% reduction in image size does not significantly affect system performance for real-time deployment. A completely different methodology is the one used by the same authors [18], which focused on using a residual neural network for crowd counting, detection of violent behavior, and classification of the density level of groups of people, demonstrating the benefits of multitasking learning. Although they manage to demonstrate such a benefit compared to learning individual tasks, they fail to overcome the methods analyzed in state-of-the-art presented in their study.

To perform crowd counting on still images from different scenes, K. Han *et al.* [14] used a convolutional neural network. They employ random Markov fields as a post-processing method, obtaining a better counting accuracy in local patches. However, the strategy of dividing the images into overlapping patches and then employing such post-processing could be a disadvantage in terms of the algorithm's execution time. A different strategy that used patches is the one exposed by H. Xiong *et al.* [19], who takes advantage of estimating the number of people in a complete image by employing the sum of the estimates in the patches that make it up. In particular, its network can generate patches in which the number of people varies in a closed interval. In this way, they trained their one-column network under a problem of closed nature. However, the network can generalize well in images where the number of people is arbitrary through the idea mentioned before. It is essential to mention that this network is considered

one of the state-of-the-art models of three of the most used datasets.

V. Sindagi and V. Patel [20] used another technique to perform crowd counting through multitasking learning. They used a high level prior to classifying the number of people in the images coarsely to make the estimation later. Thus, they can obtain high-resolution density maps using their multi-column network.

Following an approach similar to that discussed in [13], D. Sam *et al.* [21] used a three-column convolutional network, except that they use a classifier to send image patches to one of three regressors that best matches the crowd density, variations scale, perspective, and background. With this, they can vastly improve their base network at the expense of using more parameters and time to train the classifier, which is significantly challenging given the varied characteristics of crowds and their environments. In contrast, L. Zhang *et al.* [22] used a single-column network with small filters of equal size in all layers to maintain spatial resolution and create a deeper network. Notably, they perform the combination of feature maps from multiple layers to improve the robustness of the network against changes in the sizes of people's heads.

On the other hand, based on the limited amount of existing labeled data, since they are costly to obtain in terms of time and work, X. Liu *et al.* [23] tries to take advantage of unlabeled data for crowd estimation in a basic network through the learning of ranges of numbers of people. By doing so, his approach can improve the training of neural networks for the task of estimating crowds. Another improvement related to the limited amount of training data is the one devised by C. Zhang *et al.* [24]. They try to handle the model's reduced performance when analyzing unseen scenes. In particular, they examined the inference stage scenes to find the patches with the most similar scenes in the training set, with which they finely refitted their basic model. However, precisely such a non-parametric tuning scheme can take considerable time since it is necessary to find all the candidate images and patches for the fine retraining of the network. M. Reddy *et al.* [25] used a different methodology to fit the network to target scenes of a few labeled samples. They used the novel learning-to-learn approach (meta-learning) to fine-tune the network by using few labeled data, managing to overcome various methods that address this problem.

S. Aich and I. Stavness [26] focus on improving the generic object count through their heat map regulation technique. The basis of this method is back-propagating the error between a difference of predicted class activation maps and thick maps of Absolute truth Gaussian activation instead of just backpropagation of the counting error. This technique aims to reduce false positives and increase false negatives. Using such a scheme and a simple neural network, they can achieve similar performance with considerably faster inference than the model presented in [24]. On the other hand, a new type of crowd map, called Focal Inverse Distance Transform (FIDT) map, is the one devised by D. Liang *et al.* [27] and is characterized by significantly improving the individual location of people in images of dense crowds. It is worth mentioning that although his method focused on improving location, he also obtains competitive results in the task of estimating the number of people.

As can be seen, there is a clear trend in using single-column fully convolutional neural networks. In particular, a trivial architecture that meets these specifications is the encoding and decoding model, known as the autoencoder [5, 28, 29, 30, 31, 32, 33, 34]. However, an intrinsic difficulty of this model is the choice of the dimensionality of the latent coding space [35, 36]. In ideal terms, such a latent space should have the essential data characteristics that

allow the network to accurately perform the task for which it was trained. If the dimensionality of this space is less than the optimal value, the network will have severe problems executing the task. In contrast, if the dimensionality is high, the latent space will have redundant features, resulting in poor feature extraction by the network. The difficulty in finding the optimum is based on the close relationship between the latent space and the data particularities.

In conclusion, although initially, the studies focused on demonstrating the potential of crowd estimation through detection approaches and direct regression of people counting, nowadays, the density map learning approach is the most widely used. This approach is related to the extensive use of artificial neural networks, with their fully convolutional versions being the most preferred in this area. More specifically, during the last decade, there has been an increased interest in using single-column architectures, such as autoencoder models. Furthermore, given the significant improvements that FIDT maps have in locating people in dense crowds, it is estimated that researchers will use such maps in future research. Therefore, we designed a dual autoencoder architecture since there are no researches that improve localization and estimation in density and FIDT maps.

1.2 Problem Statement

In the present investigation, we address the problems of estimation and location of people in images of crowds through machine learning models. Therefore, to pose the problems, it is first mentioned that the estimation process corresponds to counting people in crowds. The location refers to where each person is located in the scenes. We carry out these tasks on a widely known dataset of images captured in the visible spectrum, in several outdoor scenes, with elevated views, varied lighting, and sparse and dense crowds.

Likewise, our machine learning models correspond to deep learning approaches, or more specifically, fully convolutional neural networks. Specifically, we used autoencoder architectures to carry out the study. In addition, we train these networks to learn the agglomerations' characteristics to generate density maps or FIDT maps. Finally, we estimate the number of people and their locations in the images from these maps.

1.3 Thesis Proposal

To reduce the complexity of obtaining an autoencoder's optimum latent space dimensionality for the simultaneous tasks of locating and estimating people in crowds, we propose a dual architecture composed of two cascaded autoencoders. The first autoencoder is responsible for generating reconstructive masking of the input images, resulting in images in which only the heads of the people are present. The second autoencoder takes the output images of the first one and generates the FIDT maps.

1.4 Hypothesis

A reconstruction task in a dual autoencoder architecture will improve localization and counting people tasks in crowd images, using density maps and Focal Inverse Distance Transform (FIDT) maps, in percentages greater than 4% and 2%, respectively, compared to a single autoencoder architecture without the reconstruction task.

1.5 Objectives

1.5.1 General Objective

The general objective of this research is to carry out a study to prove or refute the idea of improving the location and estimation of people through a reconstruction task in a specific fully convolutional neural network architecture.

1.5.2 Specific Objectives

The specific objectives that break down the general objective are presented below:

1. Implement an architecture consisting of two cascaded autoencoders.
2. Use the ShanghaiTech Part B dataset to train and test the models.
3. Train and test a single autoencoder model to generate density maps and people counting.
4. Train and test a dual network with the introduction of the reconstruction task to obtain density maps and counts.
5. Train and test the single autoencoder model to generate FIDT maps and crowd estimation.
6. Introduce the reconstruction task to train and test the dual network to obtain FIDT maps and counts.
7. Contrast the results based on standard metrics to prove or refute the hypothesis and compare the performance of the models concerning state-of-the-art methods.

1.6 Methodology

The methodology that will be applied to develop the research is set out below:

1. Implement an architecture consisting of two cascaded autoencoders.
    1.1. Theoretical, conceptual, and implementation analysis of similar architecture.

2. Use the ShanghaiTech Part B dataset to train and test the models.
   2.1. Exploratory analysis of the characteristics of the ShanghaiTech Part B dataset.
3. Train and test a single autoencoder model to generate density maps and people counting.
   3.1. Investigation of graphics card (GPU) characteristics to speed up computations.
   3.2. Analysis of the model training curves.
4. Train and test a dual network with the introduction of the reconstruction task to obtain density maps and counts.
   4.1. Observation, analysis, and contrast of training curves concerning the previous model.
   4.2. Evaluation and visual comparison of results obtained when introducing the reconstruction task in the dual architecture in contrast to the one implemented previously.
5. Train and test the single autoencoder model to generate FIDT maps and crowd estimation.
   5.1. Training analysis of the model in terms of its learning curves.
6. Introduce the reconstruction task to train and test the dual network to obtain FIDT maps and counts.
   6.1. Analysis of learning curves in the network training process.
   6.2. Evaluation and visual comparison of results concerning those obtained with the single autoencoder model.
7. Contrast the results based on standard metrics to prove or refute the hypothesis and compare the performance of the models concerning state-of-the-art methods.
   7.1. Investigation and analysis of standard metrics to contrast results.
   7.2. Analysis of results based on metrics.

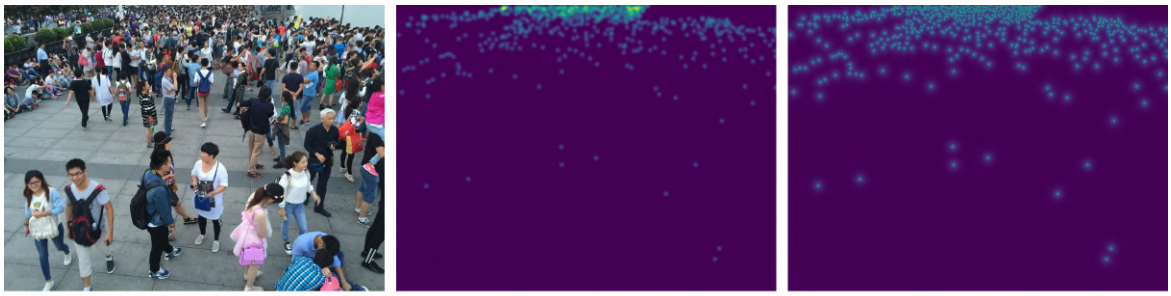
Chapter 2

Crowd Maps

2.1 Introduction

In particular, we employed two types of crowd maps: density maps and FIDT maps. These maps are created using the ShanghaiTech Part B dataset. Next, we detail step by step how to obtain each of them. It is worth mentioning that we used such maps as ground truths.

Fig. 2.1: Sample images of the ShanghaiTech Part B dataset [13, 37].



(a) Sample Image (b) Density Map (c) FIDT Map

Fig. 2.2: Sample image of the ShanghaiTech Part B dataset, density map ($\sigma = 4$, $\mu = 67$), FIDT map ($\alpha = 0.02$, $\beta = 0.75$).

2.2 ShanghaiTech Part B Dataset

We use Part B of the ShanghaiTech dataset for training and evaluation. This dataset has 400 training images and 316 evaluation images [13, 37]. The ground truths provided by this dataset are simply vectors with the position coordinates of each head in each of the images. The dataset is made up of visible spectrum images, captured in several outdoor scenes, with elevated views, varied lighting, and sparse and dense crowd densities. The images in both subsets have a resolution of 768×1024. The average number and standard deviation of people in the images of the training subset are 123 and 94. Likewise, the minimum and the maximum number of people are 12 and 576, respectively. In turn, the average, standard deviation, minimum, and maximum of people in the evaluation subset images are 124, 95, 9, and 539, respectively. The table 2.1 exposes a summary of the main characteristics of the dataset. In addition, the figure 2.1 shows a series of example images of the present dataset.

Table 2.1: Main characteristics of the ShanghaiTech Part B dataset.

Subset	Images		Number of People			
	Number	Resolution	Min	Max	Avg	Std Dev
Train Data	400	768×1024	12	576	123	94
Test Data	316	768×1024	9	539	124	95

2.3 Density Maps

A density map is a crowd map that represents people's heads by normalized Gaussian kernels [13]. The normalization aims to make the integral of each kernel equal to 1 so that we can

compute the total head count as the integral of the entire density map. In particular, it is possible to obtain this type of map employing the procedure explained below.

Consider a crowd image with N people and a set of points A containing the position (x_i, y_i), with $i = 1, 2, \ldots, N$, of each head in the image [13]. If we represent each head as a delta distribution $\delta(x - x_i, y - y_i)$, we obtain the complete image C as:

$$C(x, y) = \sum_{(x_i, y_i) \in A} \delta(x - x_i, y - y_i)\,. \tag{2.1}$$

Then, it is possible to obtain the density map as the convolution of C with a Gaussian kernel K:

$$D(x, y) = C(x, y) * K_{\sigma_i}(x, y)\,, \tag{2.2}$$

where $\sigma_i = \psi \bar{d}^i$, ψ is a constant, and $\bar{d}^i = \frac{1}{k}\sum_{j=1}^{k} d_j^i$. Y. Zhang *et al.* [13] have found empirically that $\psi = 0.3$ corresponds to the best value. The variable d_j^i is the set of distances between each head (x_i, y_i) and its k closest neighbors, $d_1^i, d_2^i, \ldots, d_k^i$. An alternative approach is to consider a constant standard deviation, which allows us to generalize the kernel size for all heads regardless of their size in the images. Here, we adopt the latter approach based on the results reported by V. K. Valloli *et al.* [30], who, with fixed-size kernels, obtained a 25% improvement in the Mean Absolute Error (MAE) metric compared to geometry-adaptive kernels using a similar architecture.

Figure 2.2(a) shows an image of crowds from the ShanghaiTech Part B dataset, whereas Figure 2.2(b) exposes its density map. Such a dataset provides crowd images, along with the location points of each head. Thus, we generated the exposed density map using the previous procedure on the set of ground truth points of the respective image. For the Gaussian kernels, we use a constant standard deviation of 4 (i.e., $\sigma_i = \sigma = 4$), and a window of 67×67 pixels (i.e., $\mu = 67$). We have tried several values for these parameters, and we selected those that generate the best results in our models.

2.4 Focal Inverse Distance Transform Maps

A Focal Inverse Distance Transform (FIDT) map is a type of crowd map characterized by accurately representing the location of each person in all kinds of crowd densities. The improvement in localization is the main difference with the density maps; however, the counting procedure requires a local maximum detection strategy [27]. Next, we explain the mathematical derivation of a FIDT map.

Consider a crowd image and a set of points A with the positions (x_i, y_i), with $i = 1, 2, \ldots, N$, of the N people's heads in the image. From this, we can obtain the Euclidean distance transform map [27] through the following expression:

$$E(x, y) = \min_{(x_i, y_i) \in A} \sqrt{(x - x_i)^2 + (y - y_i)^2}\,. \tag{2.3}$$

Then, we calculate the FIDT map as:

$$F(x, y) = \frac{1}{E(x, y)^{[\alpha \cdot E(x,y) + \beta]} + B}\,, \tag{2.4}$$

where we use $B = 1$ to avoid dividing by zero and adopt $\alpha = 0.02$ and $\beta = 0.75$ as recommended by D. Liang *et al.* [27].

Figure 2.2(c) shows the FIDT map of the sample image. We obtained this map using the previously detailed procedure on the location points provided by the dataset for each corresponding image. Comparing with Figure 2.2(b), we observe that the FIDT map significantly improves the location of people in the dense crowd area.

Chapter 3

Dual Reconstructive Autoencoder

3.1 Dual-Autoencoder Rationale

In this work, we focused our attention in designing a deep-learning model to generate both people counting and location within a given scene. From a general point of view, the use of an autoencoder would be adequate to compress the crowds' characteristics in their latent space. Nevertheless, the main problem is finding the proper dimensionality for such a coding space. Consequently, we propose to alleviate this difficulty through an architecture of two cascaded autoencoders. The first autoencoder aims to learn the characteristics of people's heads in crowd imagery in order to obtain an output image as a mosaic of circular masks of the input scene. (The center of each circle gives the location of peoples' heads in the input image.)

The second autoencoder focuses primarily on generating density maps or FIDT maps. Subsequently, we obtain the estimates of the number of people from the crowd maps. The specific objective of using our dual architecture, instead of a single autoencoder, is to separate the tasks of detecting people and generating the points representative of each head. Indeed, a single autoencoder architecture must address both tasks together, which drastically complicates training, generating even more difficulties in finding essential features for the latent space.

3.2 Proposed Architecture

We show the architecture of the proposed neural network in Figure 3.1. Here we present both variants: the one that computes the FIDT maps and the that computes the density maps, which differ only in the last block. Our model comprises two cascaded autoencoders and initially performs the reconstructive masking of the input images; for these reasons, we name it DRA. DRA models use part of the architecture proposed by V. K. Valloli *et al.* [30] as a basis.

The first autoencoder takes the crowd image and converts it to an image in which only the heads of the people are present (reconstructive masking). Subsequently, the second autoencoder takes this output and generates the FIDT map or the density map, depending on the selected

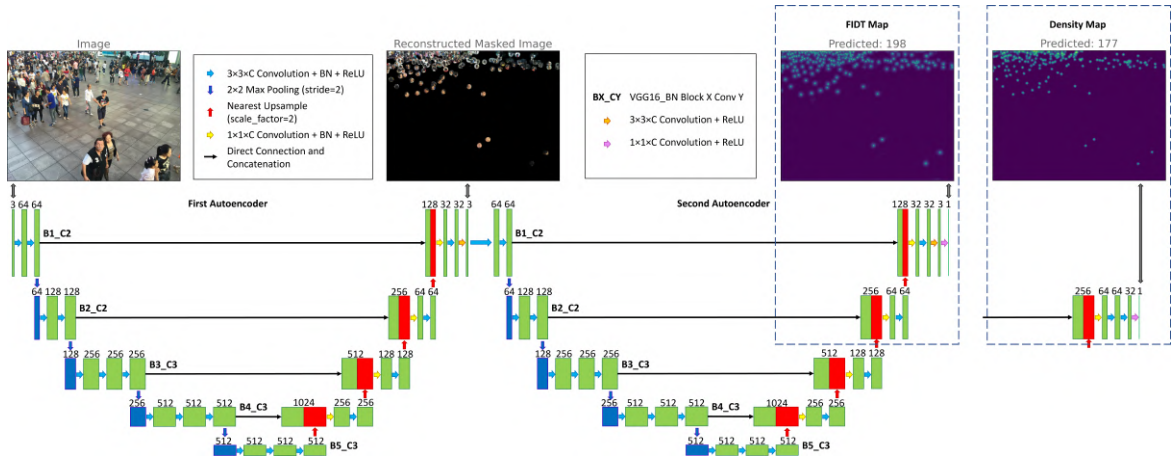


Fig. 3.1: Architecture of the Dual Reconstructive Autoencoder (DRA) model proposed in this work.

block at the end.

Both autoencoders have a contraction path and an expansion path. The variants of the DRA model (for both types of crowd maps) agree on the path of contraction, differing only at the end of the expansion path (dashed-blue square in Figure 3.1). DRA models perform feature extraction on contraction paths, which use the first five blocks of the Torchvision VGG16_BN model (without the fully connected layer). In particular, we use the pre-trained VGG16_BN on the ImageNet dataset. Next, we explain the flow of a color image in the DRA model.

3.3 Data Flow

The input RGB image is taken by the first block of the contraction path (B1_C2) of the first autoencoder, composed of two convolution layers of $3 \times 3 \times 64$ that have Batch Normalization (BN) and Rectified Linear Unit (ReLU) activation function. The feature maps are then passed through a max-pooling layer of 2×2 with stride 2, decreasing their resolution by half. The output then goes into block B2_C2, which has two convolution layers of $3 \times 3 \times 128$ with BN and ReLU. Once again, we reduce the resolution using a max-pooling layer 2×2 with stride 2 to send the feature maps to block B3_C3 composed of three convolution layers with kernels equal to those of the previous convolution layers, 256 outputs, BN, and ReLU. We apply max-pooling and three convolution layers of $3 \times 3 \times 512$, where each has batch normalization and ReLU (block B4_C3). We finish the contraction path by applying max-pooling, followed by the B5_C3 block of 3 convolutional layers with the same parameters as the previous block.

The expansion path begins applying a nearest-neighbor interpolation layer with scale factor 2, doubling the resolution of the feature maps. Then, we concatenate these maps with the outputs of block B4_C3, generating 1024 feature maps to which we apply a convolution layer of $1 \times 1 \times 256$ with BN and ReLU. Next, we send the outputs to a convolution layer with 3×3 kernel, 256 outputs, BN, and ReLU. We then double the resolution via nearest upsample,

concatenate with block B3_C3's outputs, and pass them through a $1 \times 1 \times 128$ convolution layer with batch normalization and ReLU. Next, we double the resolution, concatenate with the outputs of block B2_C2, and apply a convolution of $1 \times 1 \times 64$ followed by a convolution of $3 \times 3 \times 64$, both with BN and rectified linear unit. The expansion path ends with a further doubling of the resolution, concatenation with the outputs of block B1_C2, and application of convolutional layers of $1 \times 1 \times 32$ + BN + ReLU, $3 \times 3 \times 32$ + BN + ReLU, and $3 \times 3 \times 3$ + ReLU. The passage of the RGB image through the contraction and expansion paths of the first autoencoder generates the so-called masked reconstruction of the original image. In such an output, only the heads of the people from the original image are present.

The masked reconstruction enters the second autoencoder, passing through its contraction path first. As can be seen from Figure 3.1, such a path is identical to that of the first autoencoder. In the case of the expansion path, it is identical to its counterpart in the first autoencoder until reaching the interpolation layer of 128 feature maps. At this point lies the difference between the variant for FIDT maps and that for density maps. In the case of FIDT maps, the feature maps go through a section identical to the respective section in the expansion path of the first autoencoder until reaching the last block, which has an extra $1 \times 1 \times 1$ convolution layer + ReLU at the end, thus generating the FIDT map. On the other hand, the variant designed for density maps takes the 128 upsampled feature maps, concatenates them with those from block B2_C2, and passes them through convolutional layers of $1 \times 1 \times 64$ + BN + ReLU, $3 \times 3 \times 64$ + BN + ReLU, $3 \times 3 \times 32$ + BN + ReLU, and $1 \times 1 \times 1$ + ReLU, obtaining the density map.

As mentioned before, our model will be compared to a single autoencoder architecture that directly generates the crowd maps from the input images. For this reason, the model is called SA. The architecture of the SA model corresponds to the last autoencoder of the DRA neural network, so there is a variant for each crowd map (see Figure 3.2).

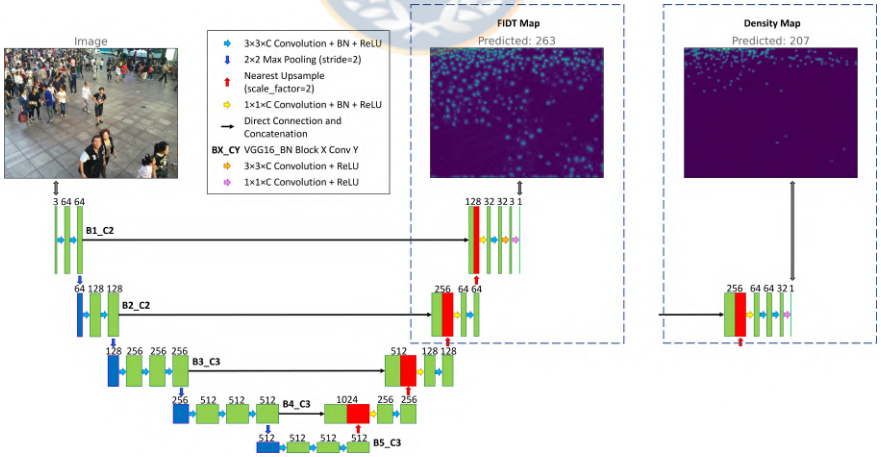


Fig. 3.2: Architecture of the Single Autoencoder (SA) model.

3.4 Training Stage

In the training of the DRA model, we use a Gaussian distribution with zero mean and 0.01 standard deviation to randomly initialize the weights of all trainable layers of the expansion paths. Moreover, we used a total of 50 epochs and a batch size of 1, generating 20,000 iterations. We also performed on-the-fly data augmentation through 14 random 400×400 crops and horizontal flips half the time. We train the first autoencoder for 8,000 iterations, the second for the following 4,000 iterations. Finally, we enable the early stopping method to monitor for validation loss when training the entire model (for both autoencoders) for the remaining 8,000 iterations.

In such a training scheme, we use different loss functions depending on the type of crowd map. In particular, the FIDT map variant uses the Mean Squared Error (MSE) and Independent Structural Similarity Index Measure (I-SSIM) loss functions, whereas the density map variant only uses MSE. Specifically, in the training of the first autoencoder of the DRA model for FIDT maps, we used the MSE loss function multiplied by a constant:

$$L_{1F} = \eta L_{MSE} = \eta \frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2 \,, \tag{3.1}$$

where $\eta = 10^5$, M is the number of pixels, y_i is the i-th ground truth value, and $\hat{y}_i$ is the i-th estimated value.

The second autoencoder of the DRA variant for FIDT maps uses a loss given by the sum of the MSE and I-SSIM losses,

$$L_{2F} = L_{MSE} + L_{I-SSIM} \,. \tag{3.2}$$

where L_{I-SSIM} has the expression [27]:

$$L_{I-SSIM} = \frac{1}{N} \sum_{i=1}^{N} L_{SSIM}(P_i, G_i) \,. \tag{3.3}$$

Here N is the total number of people, P_i and G_i are the prediction and ground truth for the i-th 30×30 independent instance region, respectively. The Structural Similarity Index Measure (SSIM) loss in (3.3) is given by:

$$L_{SSIM}(P, G) = 1 - SSIM(P, G) \,, \tag{3.4}$$

where SSIM corresponds to the Structural Similarity Index Measure, which is calculated by [27]:

$$SSIM(P, G) = \frac{(2\mu_P \mu_G + \lambda_1)(2\sigma_{PG} + \lambda_2)}{(\mu_P^2 + \mu_G^2 + \lambda_1)(\sigma_P^2 + \sigma_G^2 + \lambda_2)} \,, \tag{3.5}$$

where P and G are the predicted and ground truth maps, respectively; μ_P and σ_P (correspondingly, μ_G and σ_G) are the mean and standard deviation of predicted map, P (correspondingly of ground-truth map, G). As with the instance size, we adopt the values of λ_1 and λ_2 from [27] (i.e., $\lambda_1 = 0.0001$ and $\lambda_2 = 0.0009$). We carried out the training of the complete model through the joint loss function $L_{JF} = L_{1F} + L_{2F}$.

To train the first autoencoder of the DRA variant for density maps, we use the same loss function as for the variant for FIDT maps, that is, $L_{1D} = L_{1F}$. We used the loss function $L_{2D} = \tau L_{MSE}$, with $\tau = 10^7$, for the second autoencoder of the density map variant. As before, the joint loss L_{JD} used to train the complete model corresponds to the sum of the individual losses, namely $L_{JD} = L_{1D} + L_{2D}$.

We employed Adam optimization [38] and a learning rate equal to 10^{-4}. Likewise, for FIDT maps, we used weight decay equal to 5×10^{-4}, whereas, for density maps, we used 5×10^{-3}. All these parameters where experimentally determined to achieve the best performance. The hardware employed for training was an NVIDIA A100 Tensor Core GPU with a 40 GB HBM2 @ 1.6 TB GPU memory size, running on an accelerator-optimized (A2) Google Cloud virtual machine with 12 vCPUs and 85 GB RAM. We used Python programming language in its version 3.7.10 and PyTorch 1.9 machine learning framework.

For the first autoencoder of both variants of the DRA model, the targets were the original images multiplied by their respective binary head masks. Such masks (used only for training purposes) were obtained by thresholding the ground truth density maps (from the database) through a threshold of 10^{-5}. The targets of the second autoencoder correspond to crowd maps, where the type of map used depends on the selected DRA variant. The loss function used by each SA model variant corresponds directly to the one used by the second autoencoder of the respective DRA model variant. We train each SA variant with the same hyperparameters as the respective DRA variant to perform fair comparisons.

In order to avoid overfitting, we validated the models using 158 images out of the 316 images in the evaluation set. The validation images are only used in such a procedure and in no case to evaluate the model nor in training. In training, we used early stopping with patience equal to 2,000. This method monitored the loss functions calculated on the validation subset, for which we used a batch size equal to 7.

3.5 Evaluation Stage

We perform a patch-based evaluation for all models [30]. To explain such an approach, let us consider a single model and a single evaluation image. The procedure begins with dividing the image into nine equally-sized overlapping patches $A, B, \ldots, I$ (Figure 3.3). Later, the model is fed with the patches, generating nine inferences. We infer the complete image from the specific contribution of each of the nine small inferences, as shown in Figure 3.4. Then, to carry out the counting in a predicted density map, we integrate the entire map, whereas, in a FIDT map, it is necessary to use a procedure for detecting local maxima [27].

We employed counting, localization, and reconstruction metrics in the evaluation. The first two types of metrics are responsible for quantifying the models' performance in the tasks of counting and locating people in crowds. Specifically, localization metrics were used only for FIDT maps. Reconstruction metrics allow quantifying the DRA models' performance in the reconstructive masking task.

Among the counting metrics, we have used the Mean Absolute Error (MAE) and the Root

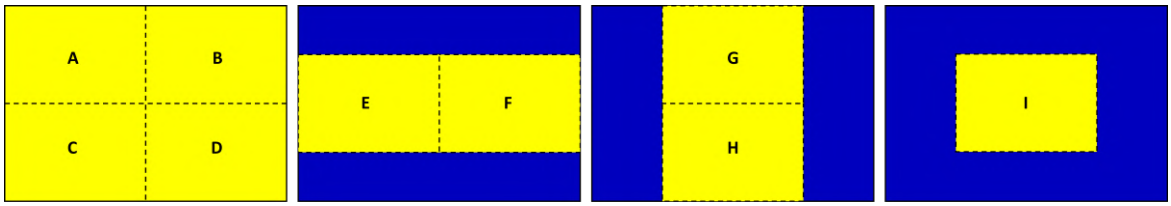


Fig. 3.3: Overlapping patches used in the evaluation stage of the models.

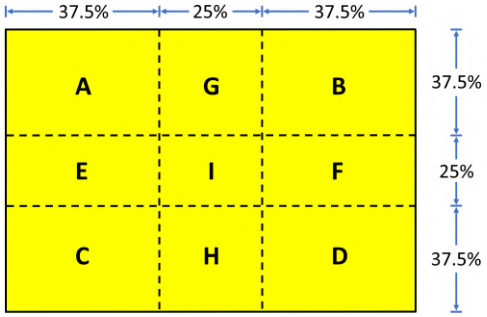


Fig. 3.4: Contribution of the nine small inferences to the complete inference of a model.

Mean Squared Error (RMSE) wich are given by:

$$MAE = \frac{1}{T}\sum_{i=1}^{T} |z_i - \hat{z}_i| \ , \tag{3.6}$$

and

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T} (z_i - \hat{z}_i)^2} \, , \tag{3.7}$$

respectively. In both metrics, z_i and $\hat{z}_i$ are the target and estimated counts, and T is the total number of evaluation images. The MAE metric measures the accuracy of the estimates, whereas RMSE measures their robustness.

The localization metrics we have employed are Precision, Recall, and F1-Score, given by:

$$\text{Precision} = \frac{TP}{TP + FP} \, , \tag{3.8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \, , \tag{3.9}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \, , \tag{3.10}$$

where TP, FP, and FN are the numbers of true positives, false positives, and false negatives, respectively. We obtained these last variables by comparing the predicted and ground truth locations, using two decision thresholds: $\sigma_1 = 4$ and $\sigma_2 = 8$. The Precision metric measures the quality of successful predictions relative to the total number of times the model predicts the

existence of an instance. In turn, Recall measures the number of correct predictions concerning the number of ground truth positives. The F1-Score metric measures the quality and quantity of correct predictions since it combines Precision and Recall.

The reconstruction metrics we have used are the RMSE, the SSIM, and the Feature Similarity Index Measure for color images (FSIMc) [39]. The FSIMc metric is defined by:

$$FSIMc = \frac{\sum_{\boldsymbol{x}\in\Omega} S_L(\boldsymbol{x})\,[S_C(\boldsymbol{x})]^{\rho}\,PC_m(\boldsymbol{x})}{\sum_{\boldsymbol{x}\in\Omega} PC_m(\boldsymbol{x})}\,, \tag{3.11}$$

where Ω is the set of all pixels in the image, $\rho = 0.03$ (based on [39]), $PC_m(\boldsymbol{x})$ is the maximum between the phase congruency of the prediction and the target, $S_L(\boldsymbol{x})$ is the similarity between the prediction and the target, whereas $S_C(\boldsymbol{x}) = S_I(\boldsymbol{x})S_Q(\boldsymbol{x})$ is the chrominance similarity. The similarities between the chromatic features $S_I(\boldsymbol{x})$ and $S_Q(\boldsymbol{x})$ are given by:

$$S_I(\boldsymbol{x}) = \frac{2I_1(\boldsymbol{x})I_2(\boldsymbol{x}) + T_3}{I_1^2(\boldsymbol{x}) + I_2^2(\boldsymbol{x}) + T_3}\,, \tag{3.12}$$

$$S_Q(\boldsymbol{x}) = \frac{2Q_1(\boldsymbol{x})Q_2(\boldsymbol{x}) + T_4}{Q_1^2(\boldsymbol{x}) + Q_2^2(\boldsymbol{x}) + T_4}\,, \tag{3.13}$$

where $T_3 = T_4 = 200$ (based on [39]), I_1 and Q_1 are the color channels of the prediction, and I_2 and Q_2 are the chrominance information of the ground truth. For an RGB image, we obtained I and Q from the following transformation [40]:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}\,, \tag{3.14}$$

where Y corresponds to the luminance information.

For the reconstruction task, the RMSE metric measures the model error, whereas SSIM and FSIMc measure the similarities in structural and chromatic terms between the predictions and the ground truths of the reconstructive masks, respectively.

We evaluated the models using the 158 images of the evaluation subset set aside exclusively for testing. None of the four neural networks have seen these images in training or validation.

Chapter 4

Results

4.1 Results

The training of the DRA model for FIDT maps begins with Figure 4.1(a), which exposes the loss L_{1F} calculated over batches of the training and validation subsets. We trained only the first autoencoder from iteration 0 to 8,000, using the mentioned loss, to perform reconstructive masking. Then, the curve has a dead zone representing a pause in the training of the first autoencoder. The beginning of the dead zone activates the training of the second autoencoder employing the loss L_{2F}. In Figure 4.1(b) we show the training curve of the second autoencoder, with the aim that it performs the generation of FIDT maps from the masked reconstructions generated by the first autoencoder. The independent training of the second autoencoder takes place from iteration 8,000 to iteration 12,000, after which we reactivated the training of the first autoencoder. Thus, we performed a joint training, where both losses operate from iteration 12,000 until the early stopping method is automatically activated. Joint training constitutes the final stage of adjustment of weights and biases to achieve a better adaptation between both parts of the network. However, as seen in Figure 4.1, most of the training of the autoencoders is done in the independent stages.

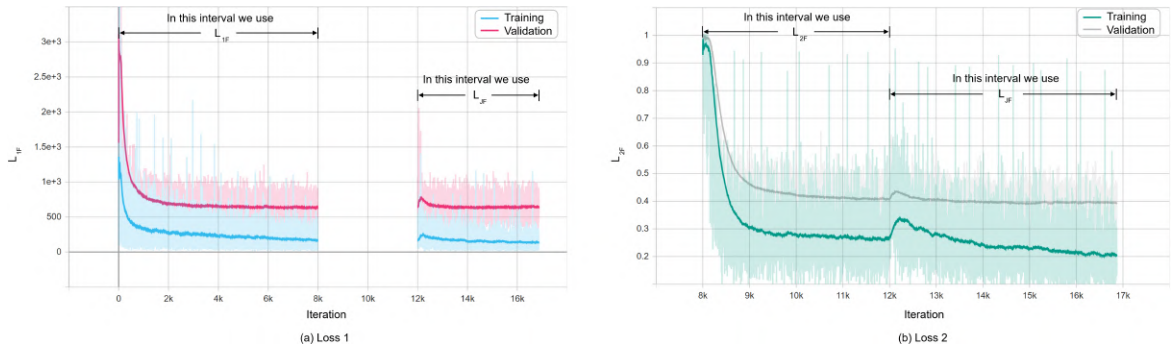


Fig. 4.1: DRA model training curves for FIDT maps.

Similarly, the DRA model variant training for density maps starts with Figure 4.2(a). Identical to the training of the previous variant, we only trained the first autoencoder during the first 8,000 iterations. In such an interval, it is possible to observe the decrease of the L_{1D} loss as the iteration increases. Subsequently, we paused the training of the first autoencoder and activated the training of the second one for the successive 4,000 iterations, using the L_{2D} loss (Figure 4.2(b)). Finally, joint training is carried out from iteration 12,000 until the early stopping method is automatically activated. Again, the most effective training of this variant occurs during the individual training of the autoencoders since, according to the curves, the joint adjustment does not provide notable improvements in learning.

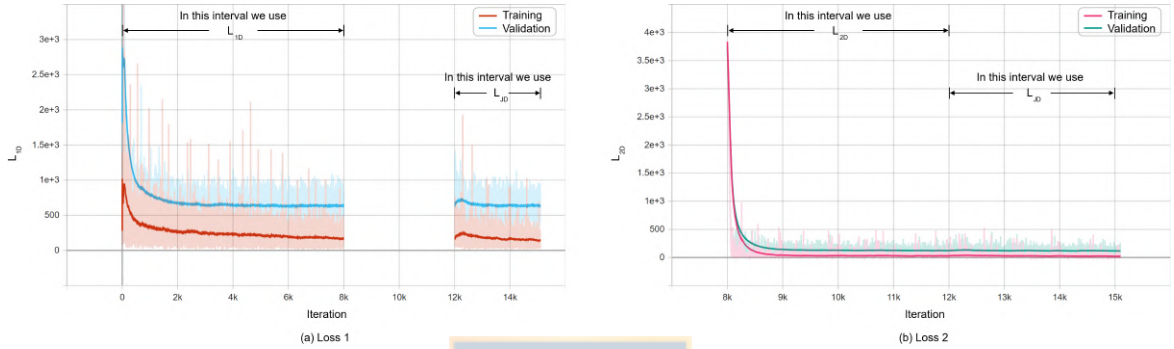


Fig. 4.2: DRA model training curves for density maps.

We display typical results from the DRA and SA models for FIDT maps in Figure 4.3. In particular, we present six images from the evaluation subset, along with the ground truth masked images, reconstructed masked images, ground truth FIDT maps, and the respective predictions of both neural networks. By comparing the predicted counts from our model (fifth column in Figure 4.3) with the predicted counts of the widely-used SA model (sixth column in Figure 4.3), one can clearly see that the DRA model significantly improves the counting and location of people in crowds compared to the SA model. The location results of this latter model (last column) differ significantly from the ground truths (fourth column), as the predicted maps have many false positives and artifacts. This result was consistent across all the images used in the evaluation stage. In the case of the FIDT maps, the counts are directly related to the locations of the people. Despite the improvement over the SA model, the DRA model's counting performance decreases for dense crowds positioned in the upper parts of the images (third sample image). However, despite this shortcoming, locating people remains competitive. Regarding the masked reconstructions made by the DRA model, it stands out that they are visually consistent with the ground truth masked images.

Figure 4.4 shows typical results from the DRA and SA neural networks for density maps. For comparison purposes, we show the same images of the evaluation subset (see Figure 4.3). In addition, the count figures of the ground truth density maps differ slightly from those of the ground truth FIDT maps due to the different mechanisms for creating the crowd maps and the different schemes for obtaining the counts. The results for the networks that generate the density maps show that the counts estimated by the SA model are closer to the actual values than the estimates generated by the DRA model when we use density maps. As expected, the

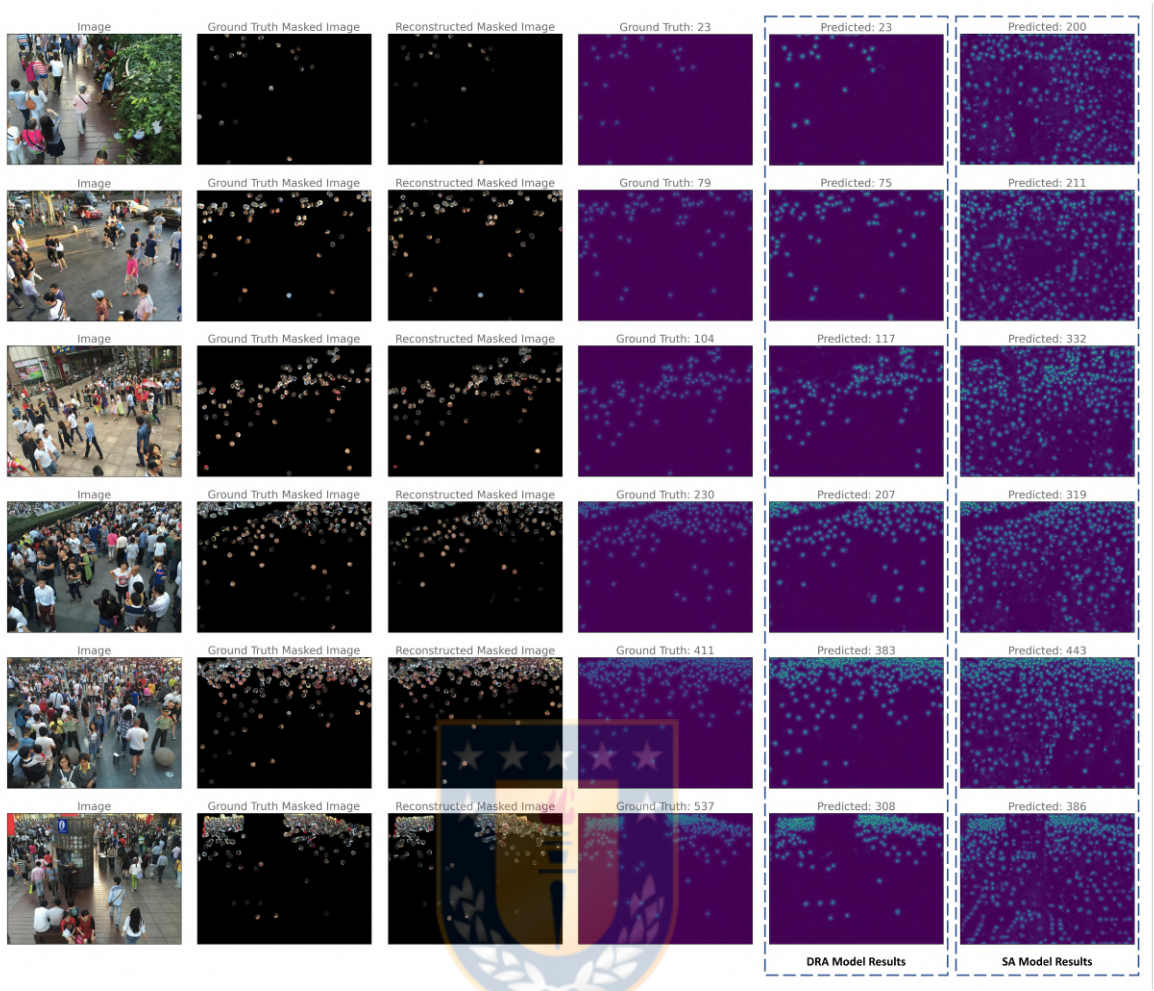


Fig. 4.3: Sample results of DRA and SA models for FIDT maps.

masked reconstructions are similar to ground truths and those generated by the DRA variant for FIDT maps.

Comparing the results of the DRA model variants for FIDT and density maps, we observe that both have similar count estimates; however, the former significantly improves the location of people in dense crowds. In turn, the SA model for density maps provides better count estimates and fewer artifacts and false positives compared to its variant for FIDT maps.

We summarize the results of the counting metrics for all models in Table 4.1. The DRA model for FIDT maps has an MAE of 13.92 and an RMSE of 31.67, whereas the SA model for the same map type obtains 121.73 and 127.61. Thus, the metrics show the considerable improvement that constitutes using the devised methodology. On the other hand, the DRA model has a lower performance for counting compared to the SA model for density maps. Likewise, the DRA variant for FIDT maps has better accuracy and robustness than its variant for density maps.

The localization metrics of the DRA and SA models for FIDT maps are summarized in Table 4.2. The Precision, Recall, and F1-Score metrics are higher in the DRA model than in

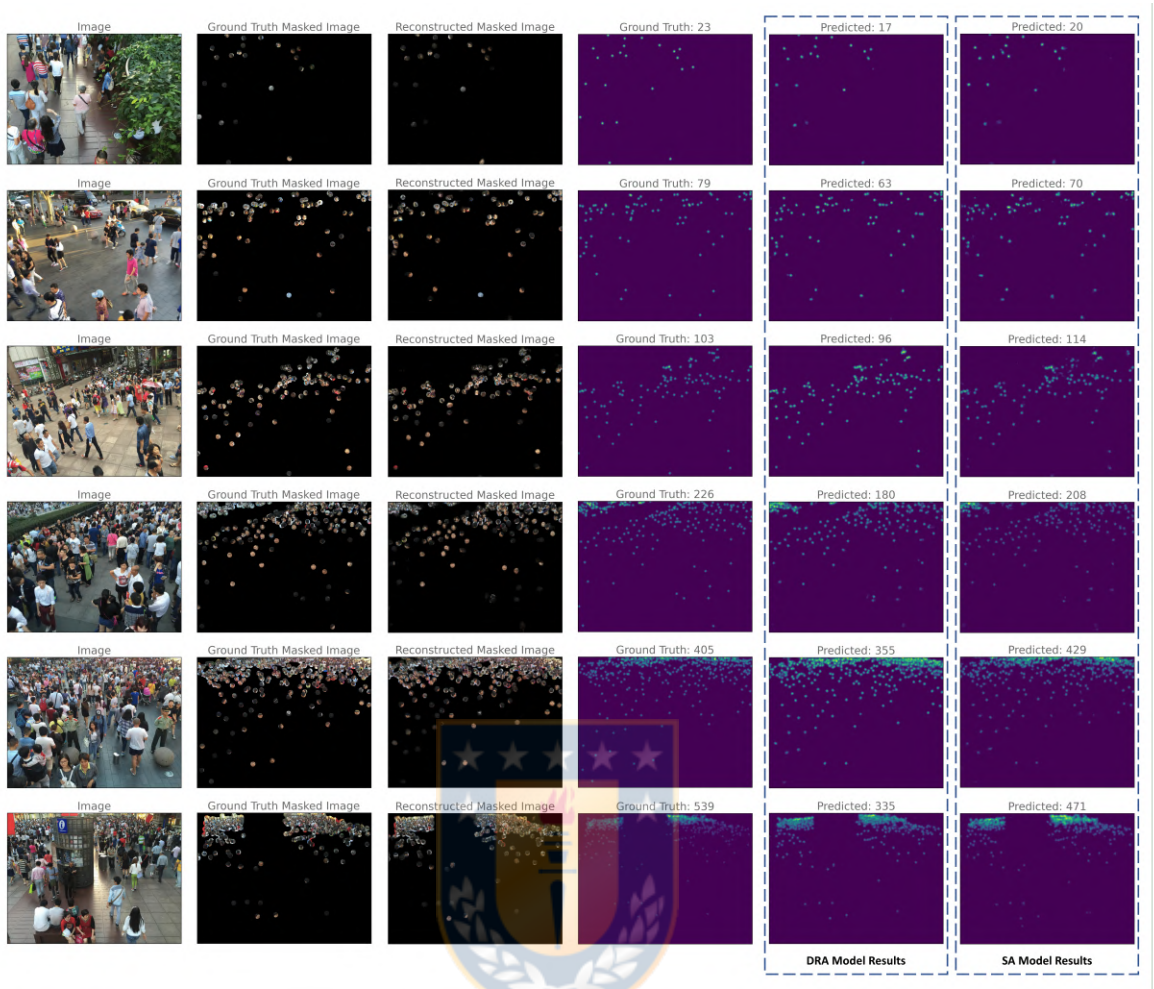


Fig. 4.4: Sample results of DRA and SA models for density maps.

the SA model, implying a significant improvement in locating people in crowds, as shown by the sample results. For example, we can observe that the proposed DRA method is able to double the Precision.

Table 4.3 shows the reconstruction metrics achieved for both variants of the DRA model. As expected, both variants have highly similar RMSE, SSIM, and FSIMc values, since they have an identical learning mechanism for their first autoencoder (number of iterations, loss functions, among others). Based on these metrics, the reconstruction performance of reconstructive masking is excellent.

Table 4.4 displays the performance of the DRA model variants and the SA model variant for density maps, along with the performance of various state-of-the-art models on the ShanghaiTech Part B dataset. It is possible to observe that our models have competitive performances. Although the SA (density maps) model performs better than the DRA (FIDT maps), it only focuses on counting, so it cannot obtain the individual location of people in dense crowds. On the other hand, unlike several state-of-the-art models, the architecture of our DRA model is simple since it corresponds to a single-column model composed of two cascaded autoencoders.

Table 4.1: Counting metrics for each variant of the DRA and SA models.

Crowd Map	Model	MAE	RMSE
FIDT map	DRA	13.92	31.67
	SA	121.73	127.61
Density map	DRA	19.87	32.59
	SA	9.50	15.57

Table 4.2: Localization metrics of the DRA and SA models for FIDT maps.

Crowd Map	Model	Precision (%)		Recall (%)		F1-Score (%)	
		$\sigma_1 = 4$	$\sigma_2 = 8$	$\sigma_1 = 4$	$\sigma_2 = 8$	$\sigma_1 = 4$	$\sigma_2 = 8$
FIDT map	DRA	63.11%	84.06%	57.61%	76.73%	60.24%	80.23%
	SA	27.11%	37.26%	54.56%	74.98%	36.22%	49.78%

Table 4.3: Reconstruction metrics for DRA model variants.

Model	Crowd Map	RMSE	SSIM	FSIMc
DRA	FIDT map	91.92	0.93	0.95
	Density map	91.96	0.93	0.94

Table 4.4: Performance of the models on the ShanghaiTech Part B dataset.

Model	Network Architecture	MAE	RMSE
C. Zhang *et al.* [24]	Basic	32.00	49.80
Y. Zhang *et al.* [13]	Multi-column	26.40	41.30
M. Marsden *et al.* [17]	Single-column	23.76	33.12
D. B. Sam *et al.* [21]	Multi-column	21.60	33.40
V. A. Sindagi *et al.* [20]	Multi-column	20.00	31.10
DRA (Density map)	**Single-column**	**19.87**	**32.59**
K. Han *et al.* [41]	Basic	17.80	26.00
L. Zeng *et al.* [16]	-	17.70	30.20
L. Zhang *et al.* [22]	Single-column	16.20	25.80
DRA (FIDT map)	**Single-column**	**13.92**	**31.67**
X. Liu *et al.* [23]	Basic	13.70	21.40
Y. Li *et al.* [14]	Single-column	10.60	16.00
SA (Density map)	**Single-column**	**9.50**	**15.57**
D. Liang *et al.* [27]	Multi-column	6.90	11.80
V. K. Valloli *et al.* [30]	**Single-column**	**6.90**	**10.30**

Chapter 5

Conclusion

5.1 General Conclusion

We conclude that our Dual Reconstructive Autoencoder (DRA) model for Focal Inverse Distance Transform (FIDT) maps generates improvements in localization and people counting compared to a single autoencoder architecture, which is widely used nowadays. For the counting task, our model decreased MAE and RMSE by 88.5% and 75.18%, respectively, compared to the metrics obtained for the Single Autoencoder (SA) model (SA model MAE: 121.73, DRA model MAE: 13.92, SA model RMSE: 127.61, DRA model RMSE: 31.67). Regarding localization metrics, respectively, for both decision thresholds, the DRA model increased the Precision by 36 (from 27.11% to 63.11%) and 46.8 (from 37.26% to 84.06%) percentage points, the Recall metric by 3.05 (from 54.56% to 57.61%) and 1.75 (from 74.98% to 76.73%) percentage points, and F1-Score by 24.02 (from 36.22% to 60.24%) and 30.45 (from 49.78% to 80.23%) percentage points. Although the computational cost and inference time of the DRA model are approximately twice that of the SA network, our neural network widely exceeds the hypothesized improvement percentages for counting and location tasks, which justifies its use and therefore validate the hypothesis of improvement for FIDT maps. These improvements were achieved due to the proposed architecture we have designed and the methodology of separating the tasks of detecting people and generating points representative of each head into independent autoencoders. Our neural network obtains crowd estimates similar to those of state-of-the-art models, accurate locations for all crowd density types, and excellent masked reconstructions. Despite this, our task-division approach failed to improve counting performance when density maps were used compared to the SA model. Indeed, the SA neural network outperformed the DRA model by 47.81% for the MAE metric (SA model MAE: 9.50, DRA model MAE: 19.87) and 47.77% for the RMSE metric (SA model RMSE: 15.57, DRA model RMSE: 32.59), which refutes the hypothesis of improvement for density maps. However, the SA model only focuses on counting people, unable to obtain individual locations of people in dense crowds, whereas our model can generate both due to the dual architecture.

5.2 Future Works

Among the future works, we will implement the DRA neural network for FIDT maps in the facilities of the campus of the Universidad de Concepción, Chile. We will deploy visible, near-infrared, and long-wave-infrared security cameras to characterize crowds using the proposed dual-autoencoder approach. Moreover, we will use a standalone 5G mobile communications network for centralized communication between the cameras and a deep learning server. In addition, we will generate an assembly of neural networks using the intermediate output of the DRA model to feed a facial expression recognition model. In this way, we will achieve a prototype of an intelligent, accurate, robust, fast, and effective Earthquake Early Warning System (EEWS) to help authorities make complex decisions at critical moments triggered by natural disasters.

References

[1] "National situation of COVID-19 in Chile," *Government of Chile*. [Online]. Available: https://www.gob.cl/pasoapaso/cifrasoficiales/, Accessed on: Sep. 13, 2022.

[2] "WHO Coronavirus (COVID-19) dashboard," *World Health Organization*. [Online]. Available: https://covid19.who.int/, Accessed on: Sep. 13, 2022.

[3] "WorldRiskReport 2022," *Bündnis Entwicklung Hilft and Ruhr University Bochum*. [Online]. Available: www.WorldRiskReport.org, Accessed on: Sep. 13, 2022.

[4] G. P. Hayes, E. K. Myers, J. W. Dewey, R. W. Briggs, P. S. Earle, H. M. Benz, G. M. Smoczyk, H. E. Flamme, W. D. Barnhart, R. D. Gold, and K. P. Furlong, "Tectonic summaries of magnitude 7 and greater earthquakes from 2000 to 2015," *U.S. Geological Survey*, 2017, https://doi.org/10.3133/ofr20161192.

[5] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "CNN-based density estimation and crowd counting: A survey," *arXiv preprint*, arXiv:2003.12783, 2020.

[6] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.

[8] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 794-801, doi: 10.1109/CVPR.2009.5206638.

[9] B. Benfold and I. Reid, "Stable multi-target tracking in real-time surveillance video," *CVPR 2011*, 2011, pp. 3457-3464, doi: 10.1109/CVPR.2011.5995667.

[10] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *BMVC*, volume 1, page 3, 2012.

[11] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013, pp. 2547-2554, doi: 10.1109/CVPR.2013.329.

[12] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia - MM '15*, 2015, doi: 10.1145/2733373.2806337.

[13] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 589-597, doi: 10.1109/CVPR.2016.70.

[14] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[15] L. Boominathan, S. Kruthiventi, and R. Babu, "CrowdNet: A deep convolutional network for dense crowd counting," *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, doi: 10.1145/2964284.2967300, 2016.

[16] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, 2017, pp. 465-469, doi: 10.1109/ICIP.2017.8296324.

[17] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2017, doi: 10.5220/0006097300270033.

[18] M. Marsden, K. McGuinness, S. Little, and N. E. O'Connor, "ResnetCrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, 2017, pp. 1-7, doi: 10.1109/AVSS.2017.8078482.

[19] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, "From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, doi: 10.1109/iccv.2019.00845.

[20] V. A. Sindagi and V. M. Patel, "CNN-Based cascaded multi-task learning of high-level prior and density estimation for crowd counting," *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, 2017, pp. 1-6, doi: 10.1109/AVSS.2017.8078491.

[21] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, 2018, pp. 1113-1121, doi: 10.1109/WACV.2018.00127.

[23] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, doi: 10.1109/cvpr.2018.00799.

[24] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 833-841, doi: 10.1109/CVPR.2015.7298684.

[25] M. Reddy, M. Asiful Hossain, M. Rochan, and Y. Wang, "Few-Shot Scene Adaptive Crowd Counting Using Meta-Learning," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, doi: 10.1109/wacv45572.2020.9093409.

[26] S. Aich and I. Stavness. "Improving object counting with heatmap regulation," *CoRR*, abs/1803.05494, 2018.

[27] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization and counting in dense crowd," *arXiv preprint*, arXiv:2102.07925, 2021.

[28] M. Liu, J. Jiang, Z. Guo, Z. Wang, and Y. Liu, "Crowd counting with fully convolutional neural network," *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 953-957, doi: 10.1109/ICIP.2018.8451787.

[29] D. B. Sam, N. N. Sajjan, H. Maurya, and R. V. Babu, "Almost unsupervised learning for dense crowd counting," in *AAAI*, 2019.

[30] V. K. Valloli and K. Mehta, "W-Net: Reinforced u-net for density map estimation," *arXiv preprint*, arXiv:1903.11249, 2019.

[31] H. Bai and S.-H. Chan, "CNN-based single image crowd counting: Network design, loss function and supervisory signal," *arXiv preprint*, arXiv:2012.15685, 2020.

[32] B. Li, H. Huang, A. Zhang, P. Liu, and C. Liu, "Approaches on crowd counting and density estimation: A review," *Pattern Anal Appl*, Feb. 2021, doi: 10.1007/s10044-021-00959-z.

[33] N. Ilyas, A. Shahzad, and K. Kim, "Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation," *Sensors*, vol. 20, no. 1, p. 43, Dec. 2019, doi: 10.3390/s20010043.

[34] V. Nguyen and T. D. Ngo, "Single-image crowd counting: A comparative survey on deep learning-based approaches," *Int J Multimed Info Retr*, vol. 9, no. 2, pp. 63–80, Oct. 2019, doi: 10.1007/s13735-019-00181-y.

[35] Y. Liu, E. Jun, Q. Li, and J. Heer, "Latent Space Cartography: Visual Analysis of Vector Space Embeddings," *Computer Graphics Forum*, vol. 38, no. 3. Wiley, pp. 67–78, Jun. 2019. doi: 10.1111/cgf.13672.

[36] Z. Li, R. Tao, J. Wang, F. Li, H. Niu, M. Yue, and B. Li, "Interpreting the Latent Space of GANs via Measuring Decoupling," in *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 1, pp. 58-70, Feb. 2021, doi: 10.1109/TAI.2021.3071642.

[37] "ShanghaiTech dataset," *Kaggle*. [Online]. Available: https://bit.ly/3Lr4w6W, Accessed on: Sep. 13, 2022.

[38] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv*, 2014, doi: 10.48550/arXiv.1412.6980.

[39] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," in *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378-2386, Aug. 2011, doi: 10.1109/TIP.2011.2109730.

[40] C. Yang and S. H. Kwok, "Efficient gamut clipping for color image processing using LHS and YIQ," *Optical Engineering*, vol. 42, n°3, pp. 701-711, Mar. 2003, doi: 10.1117/1.1544479.

[41] K. Han, W. Wan, H. Yao, and L. Hou, "Image crowd counting using convolutional neural network and markov random field," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 21, no. 4, pp. 632–638, Jul. 2017.

[42] F. Lamas, K. Duguet, J. E. Pezoa, G. A. Montalva, S. N. Torres, and W. Meng, "Crowd detection and estimation for an earthquake early warning system using deep learning," *Proc. SPIE 12101 Pattern Recognition and Tracking XXXIII*, May 2022, doi: 10.1117/12.2622392.

[43] F. I. Lamas, J. E. Pezoa, S. E. Godoy, G. A. Saavedra, S. N. Torres, G. A. Montalva, and W. Meng, "Dual Reconstructive Autoencoder for Crowd Localization and Estimation in Density and FIDT Maps," submitted to *IEEE Access*, Sep. 2022.