Investigating the Performance of Computer Vision Models When Augmented with Artificially Generated Training Data

RQ: To what extent is it viable to use synthetically generated data to train a computer vision model for object counting and localization?

TuKoKe 2024

Computer Science

Word count: 4000

Contents

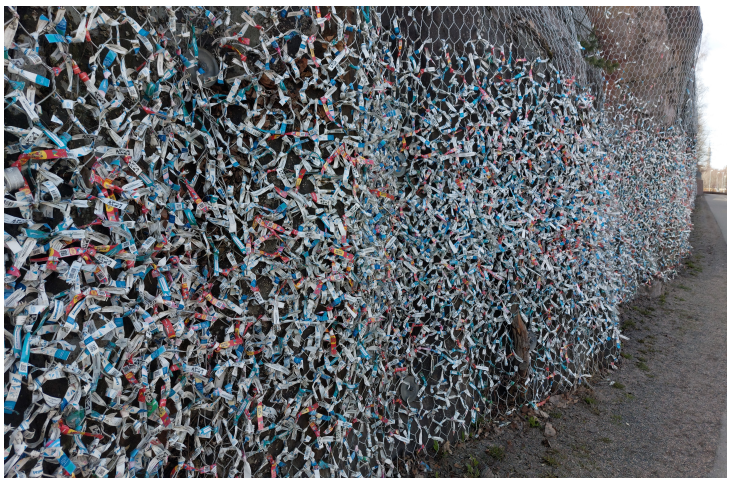

Figure 1: The wall of tickets.

1 Research question

The idea for this investigation surfaced upon passing a mesh fence to which countless tickets from a nearby attraction park have been tied, as shown in Figure 1. The number was so overwhelming, that it wouldn't realistically be possible to quantify them by hand. An algorithmic[1] approach would likely fail too due to occlusion and significant variance in the demeanor of the tickets. Finally, an approach involving Artificial Intelligence (AI) had the possibility of succeeding but would require substantial amounts of data to train the model on that was not readily available. This is a broader issue in the growing field of machine learning. As training data becomes increasingly valuable to firms racing to build ever potent neural networks, it continues to be expensive to manually collect and annotate large datasets. Combining personal experience with the broader context naturally leads to the following research question: *To what extent is it viable to use synthetically generated data to train a computer vision model for object counting and localization?*

2 Introduction

Synthetic data generation is a growing interest in the scientific community, with claims of it being "one of the most promising general techniques on the rise in modern deep learning" [13]. Recently, researchers at Stanford have explored the possibility of extending the 'LIVECell' dataset by artificial replicas [15]. Furthermore, a team at MIT has shown

[1]For instance, using Canny edge detection and a maximum filter [12].

that a model trained to classify human actions (such as waving or walking) from synthesized video clips not only circumvented privacy considerations, but also outperformed its naturally-trained counterpart [18].

In addition to the contributions made, this paper seeks to extend the research on synthetic data particularly in relation to object counting and localization with more diverse datasets. To do so, three unrelated, natural datasets with different sizes and complexities were chosen. Next, an algorithm to produce synthetic replicas was developed based on the parameters of real samples. Finally, a neural network was trained on different proportions of natural to synthetic data, and its ability to quantify the number of objects and identify their positions was evaluated. Overall, it was found that in cases where natural data was incomplete, synthetic data could provide substantial performance improvements. However, otherwise or in high concentrations, it could also hinder the model by overly 'diluting' the training set.

3 Background Information

3.1 Computer vision

Computer vision is a field of AI that enables computers to derive meaningful information from digital images, video and other visual inputs [10]. Nowadays, it is becoming ever-present in applications ranging from image-to-text conversions to self-driving vehicles.

As the field grows, computer vision remains reliant on two underlying technologies: deep learning and convolutional neural networks (CNNs) [10]. Deep learning, a subset of machine learning, refers to the attempt to simulate the learning process of the human brain by a digital model. It employs numerous layers of neurons (similar to those in the brain) placed back-to-back, making it possible for the network of neurons to create meaningful connections. Provided a significant amount of training data, this allows the model to connect features such as dog ears with the concept of a dog.

A convolutional neural network aids the model in interpreting the image. Rather than attempting to process it at all once, the convolutional approach involves analysing the input in several iterations by breaking it down into smaller chunks, with help of a

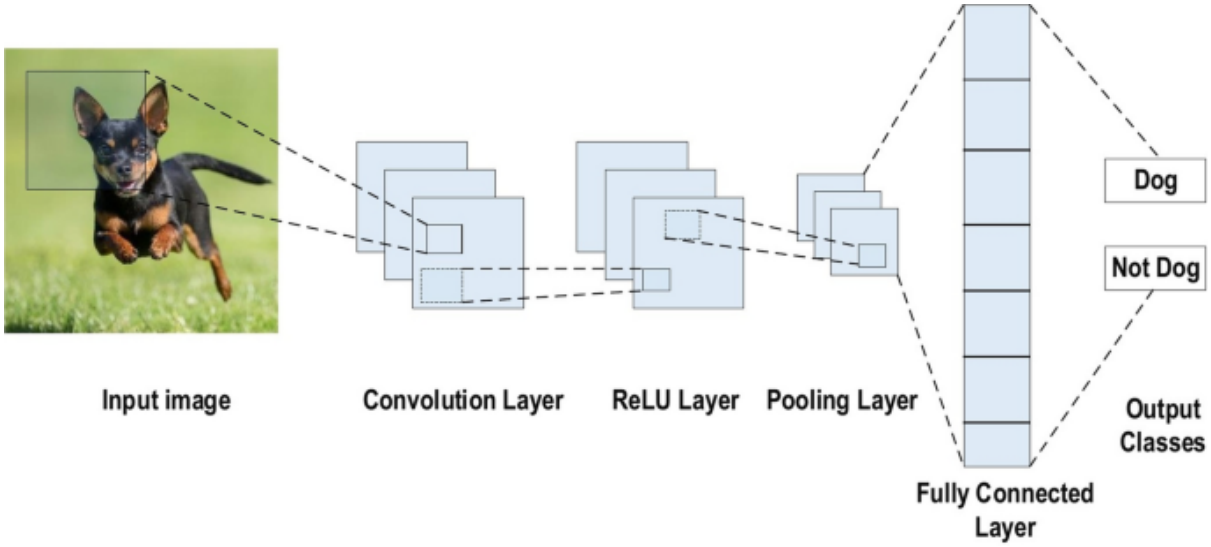

Figure 2: A neural network that takes an image as its input, subdivides it into smaller regions by convolution, and performs deep learning. [20]

kernel/filter – a matrix of numbers that isolates smaller segments of the image. This is what allows the model to extract both global (e.g. edges, colors) and specific (e.g. dog ears) features from an image. An example of a CNN that follows the deep learning pattern is visualized in Figure 2.

3.2 Object counting

Object counting is a subset of computer vision that involves taking a picture and quantifying the number of times a certain feature is present. This problem is growing increasingly prevalent in applications such as crowd management, quality assurance in manufacturing, and cell counting in microbiology [5].

Traditionally, there are three possible approaches in designing a model for this purpose [11]. The first relies on using a CNN that has been crafted for object detection. In other words, the model is already able to localize the positions of certain features in an image. The output of such a network is commonly formatted as 'bounding boxes', visualized in Figure 3. It then becomes just a matter of counting the number of these detections. However, this approach suffers in accuracy when it comes to densely populated, low-resolution images, especially those riddled by occlusion, which occurs when objects overlap each other.

In contrast, a regression-based approach involves directly mapping from the extracted image features to the count of objects. In other words, such models circumvent the step

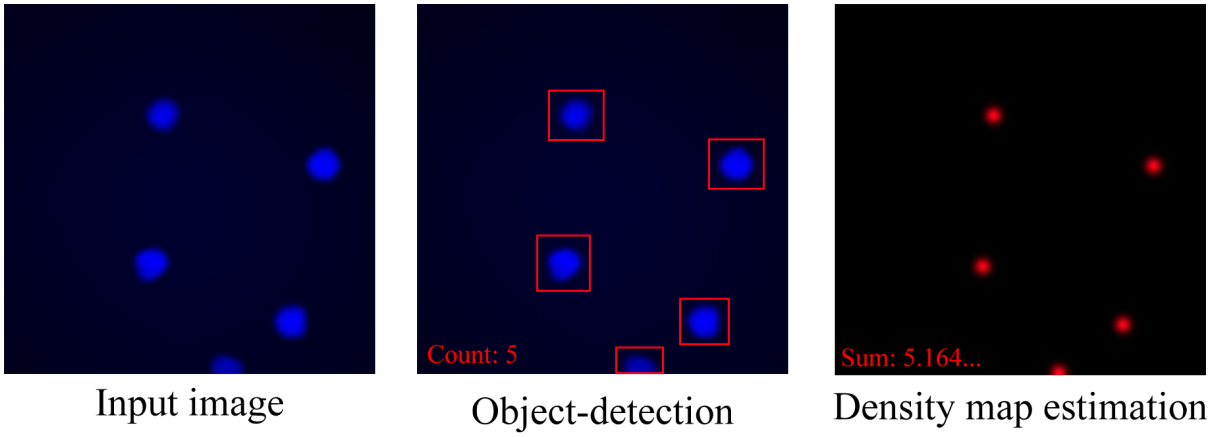

Figure 3: The differences in the outputs produced by two models constructed using the detection-based vs density-based approaches.

of localization (determining the positions of features) and directly produce a numeric output for their count. This is an effective technique, but it precludes the ability to retrieve information about the distribution of objects in an image and their locations, which can prove valuable in pragmatic contexts [11].

In such situations, density map estimation is the preferred option [11]. The approach relies on training a model to produce an output image that contains 'hotspots' representing the positions of detected features, as shown in Figure 3. This method manages to bypass the negatives of detection-based counting by deriving estimates from low-level features (as opposed an object as a whole), but also retains location information absent from regression-based counting. Furthermore, finding the number of objects of interest becomes a matter of simply taking the integral (sum) of the density map in a region of interest.

3.3 Loss function

Much like a human being, a neural network requires continuous feedback in order to improve. In machine learning, this is described by a loss function that attempts to numerically quantify how far off the model is from ground-truth and steer it in the right direction. There are a variety of different loss functions in use, but this paper considers Mean Absolute Error (MAE) for its simplicity and computational ease [5]. Mathematically, the function is defined as follows,

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|t_i - p_i| \ , \tag{1}$$

where t_i is the ground-truth count and p_i is the predicted number of objects for the i-th sample, out of N total samples.

Breaking this down results in a fairly intuitive formula. First, find how far off the predicted value is from ground-truth $(t_i - p_i)$. Then, take the absolute value of that, since the model could be either under- or overcounting. Finally, perform these steps for all samples in a dataset and take the arithmetic mean.

3.4 Evaluation metrics

One way to evaluate the accuracy of density-based models is by using the loss function described above. However, MAE takes into account solely the error in the *count* of objects detected and so neglects the localization data provided by the density map. To abet this, [9] and [16] propose approaches that are inspired from object detection.

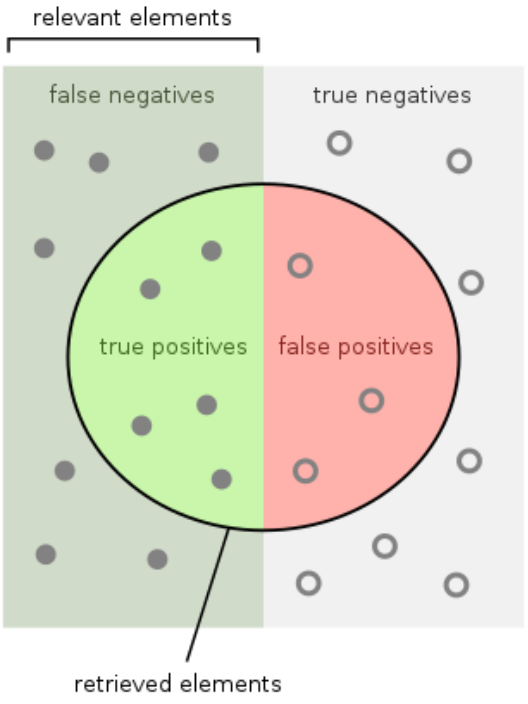

Figure 4: The classifications that can occur for a system designed to retrieve filled gray dots. [7]

In object detection, there are typically two metrics applied to assessment: precision and recall. These require an underlying understanding of four outcomes that such a model could produce: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). Consider a neural network trained to detect full gray dots but not hollow gray dots, as shown in Figure 4. A TP would mean that the model correctly identified the position of the desired dot. Conversely, a FP occurs when the neural network incorrectly detected an object, in this case a hollow dot. Then, FNs are dots that the model should have identi-

fied but failed to. Finally, a TN is a dot that should not be and was not detected by the network.

From here, precision can be defined as follows,

$$\text{“Precision”} = \frac{\text{“Correct predictions”}}{\text{“Total predictions”}} = \frac{\text{“True positives (TP)”}}{\text{“TP + False positives (FP)”}} \ , \tag{P}$$

Conversely,

$$\text{”Recall”} = \frac{\text{”Correct predictions”}}{\text{”Total ground-truth”}} = \frac{\text{”TP”}}{\text{”TP + False negatives (FN)”}} \ , \tag{R}$$

Intuitively, precision measures the accuracy of a model, or how correct the model is in its predictions of features. On the other hand, recall tells us how good the model is at finding these features. Note that both metrics can be expressed as percentages since their range is $[0, 1]$.

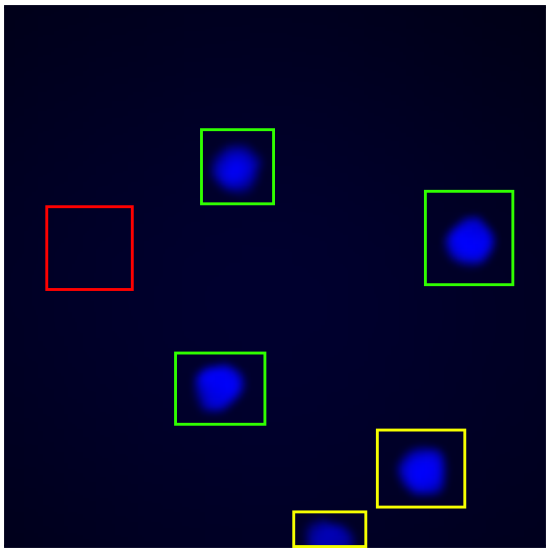

Figure 5: Classified output from an object detection model.

To illustrate this, consider a sample output from an object detection model in Figure 5, where a green bounding box corresponds to a TP, red to a FP, and yellow to a FN. In this case,

$$\text{”Precision”} = \frac{3}{3+1} = \frac{3}{4} = 75\%$$

$$\text{”Recall”} = \frac{3}{3+2} = \frac{3}{5} = 60\%$$

In practice, precision and recall are often combined to better describe the performance of a neural network. A straightforward way to do so would involve taking their arithmetic mean. However, the more widely used metric is derived by taking their *harmonic*[2] mean [8]. This is known as the F1-score and can be expressed as follows,

$$F_1 = 2 \times \frac{\text{“precision”} \times \text{“recall”}}{\text{“precision”} + \text{“recall”}} \tag{F1}$$

[2]The harmonic mean offers several mathematical advantages over the arithmetic mean [8]

The equation may appear unintuitive, but at its core it simply provides a way to compile precision and recall into a single value. Crucially, the range of the F1-score remains $[0, 1]$, meaning that it can also be expressed as a percentage. Continuing from the example above,

$$F_1 = 2 \times \frac{0.75 \times 0.6}{0.75 + 0.6} = 0.\overline{66} \approx 66.7\%$$

3.5 Data augmentation

Finally, the last component in a successful neural network is an abundance of training data. This is often one of the highest hurdles, as manual data collection and annotation can easily grow to be laborious and costly. To resolve this dilemma, data is often augmented. In practice, data augmentation involves applying random transformations (such as rescaling, rotation, contrast etc.) to original images. This is done so that the neural network can grow used to more diverse circumstances, i.e. avoid overfitting, which occurs when the model becomes overly specialized on particular data and as such loses accuracy in a more general case [14]. Nevertheless, data augmentation is, to an extent, limited by the size and diversity of the original training set. For instance, a training set that is biased towards depictions of blue cells is not trivial to augment to better represent red cells.

3.6 Data synthesis

Synthetic data is data that has been artificially produced in an attempt to mimic its real-world counterpart. Besides filling gaps in naturally acquired datasets, one of the motivations of using synthetic data is the possibility to generate private data that cannot be traced back to individuals, which can be particularly important in sensitive fields like medicine or banking [19]. Furthermore, synthetic data offers scalability that cannot be rivaled by manually annotated datasets, even with augmentation. This is rapidly gaining significance as the most recent advances in machine learning are fueled by feeding exponentially greater amounts of data to models [4].

At the same time, synthetic data does not come without drawbacks. Often, the process of generating artificial data demands an upfront cost to develop a system capable

of synthesizing replicas that can sufficiently mimic real-world samples. In practice, such systems can range from a simple algorithm to a separate neural network that is solely responsible for generating training data for another network [13].

4 Hypothesis

It is generally held that the quantity of training data is positively correlated with the model's performance [15]. At the same time, it is recognized that synthetic data will likely not achieve the quality[3] of the original training set. Thus, the contention of this paper is that at equal numbers, purely natural data will outperform purely synthetically generated equivalents. However, upon augmenting real data with synthetic images, the performance will increase linearly due to more total data being available to the model.

5 Methodology

At its core, three procedures were involved in obtaining the experimental results for this study.

1. Data from natural datasets was loaded and preprocessed into a usable format.
2. Synthetic data was generated based on the parameters of its natural counterpart.
3. A neural network was trained on datasets with different ratios of synthetic data to natural data.

5.1 Natural data

Three datasets of different properties were chosen in an attempt to provide a diverse spectrum for experimentation. The sizes of the datasets (in number of images) decrease, while their relative visual complexity increases, in the order of their introduction below.

5.1.1 Cells

The first of the datasets used is the 'VGG Cells' dataset, taken from [2]. It consists of 200 colored (RGB) images that depict fluorescent cells, with dimensions of 256x256

[3]Quality, in this context, is used to refer to how well the training data corresponds to the validation data.

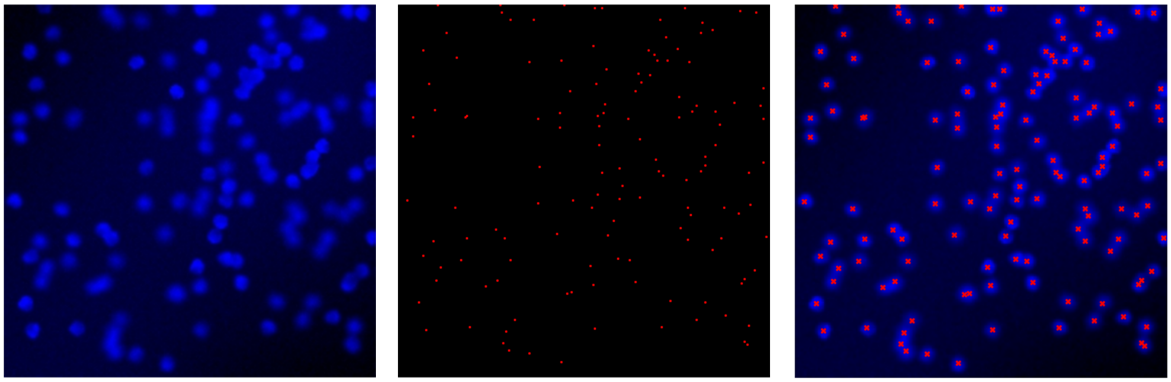

Figure 6: Input and annotation images provided by the 'cells' dataset. The image on the right is derived by placing red crosses in the places of annotations for visualization.

pixels. Each image is annotated by another image of identical dimensions that contains red dots in the corresponding positions, as shown in Figure 6. Lastly, it should be noted that this dataset is actually synthetically generated itself by the system proposed in [1]. Nevertheless, it very closely mimics reality and is widely regarded as the standard when it comes to benchmarking cell counting models [2].

5.1.2 Blueberries

Figure 7: An example image from the 'blueberry' dataset. Its annotation has been overlaid for visual ease in a similar format to the one presented in Figure 6.

The second dataset, 'Deep NIR', is adapted from [17]. As visualized in Figure 7, it includes over 4000 images of 11 different classes of fruit, annotated with bounding boxes according to the YOLO format [17]. For the purposes of this study, the dataset was pruned to only contain 78 RGB images of blueberries. Furthermore, the annotations were converted from the aforementioned format to images with a white, 1x1 pixel dot in the center of each previously defined bounding box, similar to the structure of the 'cells' dataset. Next, all image files were resized (and padded if necessary) to have dimensions of 256 by 256 pixels. The purpose of including this dataset

in the experimentation is to explore the possibility of synthesizing data for more complex cases than those covered by the 'cells' dataset. In particular, the 'blueberry' dataset includes more frequent and problematic occlusions, foreign objects and other noise.

5.1.3 Tickets

Finally, the third dataset was manually compiled for this study. As visualized in Figure 8, it consists of 49 RGB, 512x512 pixel images of wristband tickets, annotated by another image with 1x1px red dots at their 'knots'. This dataset features perhaps the most severe instances of occlusion and complex objects, which also makes it labour-intensive and error-prone to manually annotate. Thus, it's a valuable asset for analysis as a synthetic approach could provide not only more, but potentially more accurate training data.

Figure 8: An example image from the 'tickets' dataset.

5.2 Synthetic data

Synthetic training data was generated for each dataset with intent to mimic it as closely as possible on two levels: statistical and visual.

Statistically, the synthetic datasets were tailored to follow the arithmetic mean and variance of the counts of objects in the original *validation* sets. In other words, this ensures that the artificial images contain on average a similar amount of objects as their natural counterparts.

On a visual level, the datasets were replicated by extracting a range of objects from original images and then 'pasting' them into synthetic images, with respect to the aforementioned computed distribution. Furthermore, to prevent overfitting (the model becoming overly specialized on synthetic rather than natural data), randomized transformations,

such as rotation and blur, were applied.

5.2.1 Cells

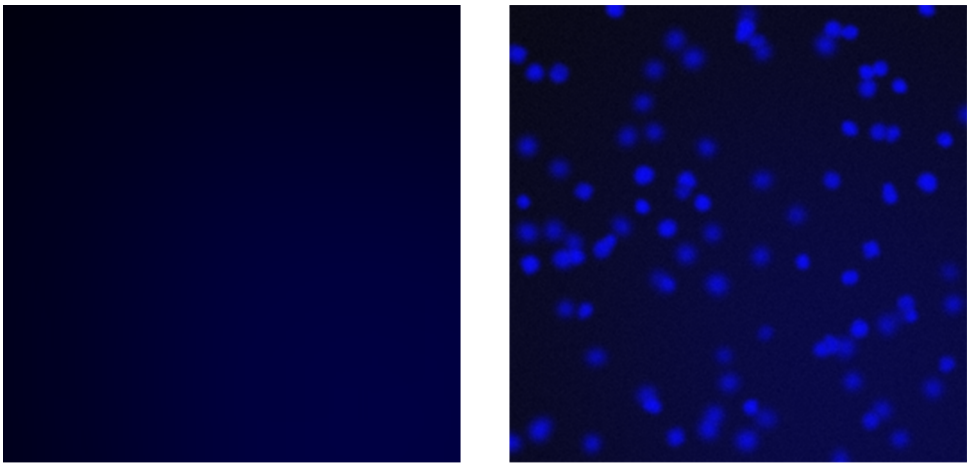

Figure 9: The process for generating a single synthetic image for the 'cells' dataset.

First, 71 unique, arbitrary cells were selected from the natural images to use as samples. Then, for a single replica image, a black background was created, as seen in Figure 9. Next, blue ellipses were overlaid and blurred in an attempt to mimic the background glow visible in original images. Then, a statistically appropriate number of cells was pasted over random locations on the previously synthesized background image, drawing (possibly duplicates) from the 71 extracted samples. Simultaneously, blurring was applied to blend the cells into the background.

5.2.2 Blueberries

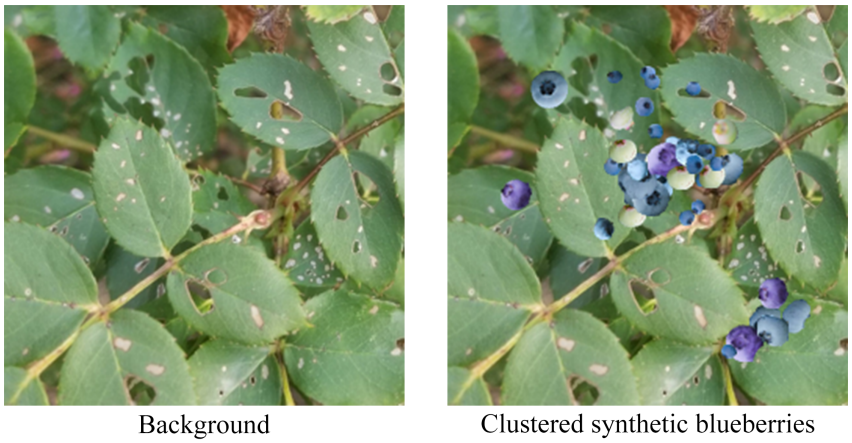

Figure 10: The process for generating a single synthetic image for the 'blueberry' dataset.

Again, the first step involved extracting 35 arbitrary samples of blueberries from the original dataset. Then, for a single image, a background was selected from a list of 7 forest backgrounds and randomly rotated and/or cropped for diversity. Blueberries were then placed in clusters over the background in an attempt to mimic their natural growing pattern, as shown in Figure 10.

5.2.3 Tickets

Figure 11: The process for generating a single synthetic image for the 'tickets' dataset.

The approach employed for generating synthetic images of tickets closely mimics the two above. A randomly cropped background was generated from a set of 22 background images. Then, ticket samples were randomly rotated and positioned over it, drawing from 37 cut-outs from the original data.

5.3 Model architecture

The neural network architecture chosen for this study is a widely used CNN known as U-Net, first introduced in [3]. Fundamentally, ***U***-Net earns its name for the way the model architecture is constructed, as shown in Figure 12. In a nutshell, the key to the network's performance lies in its ability to extract features from multiple levels of the image (moving down the U-shape), and then to combine the findings (moving up and across the U-shape). This is also what allows the architecture to be used for density map creation, as the output is an image of the same dimensions as the input. In practice, there exist numerous ways to implement and modify U-Net, but for this paper, the implementation was adapted from [5].

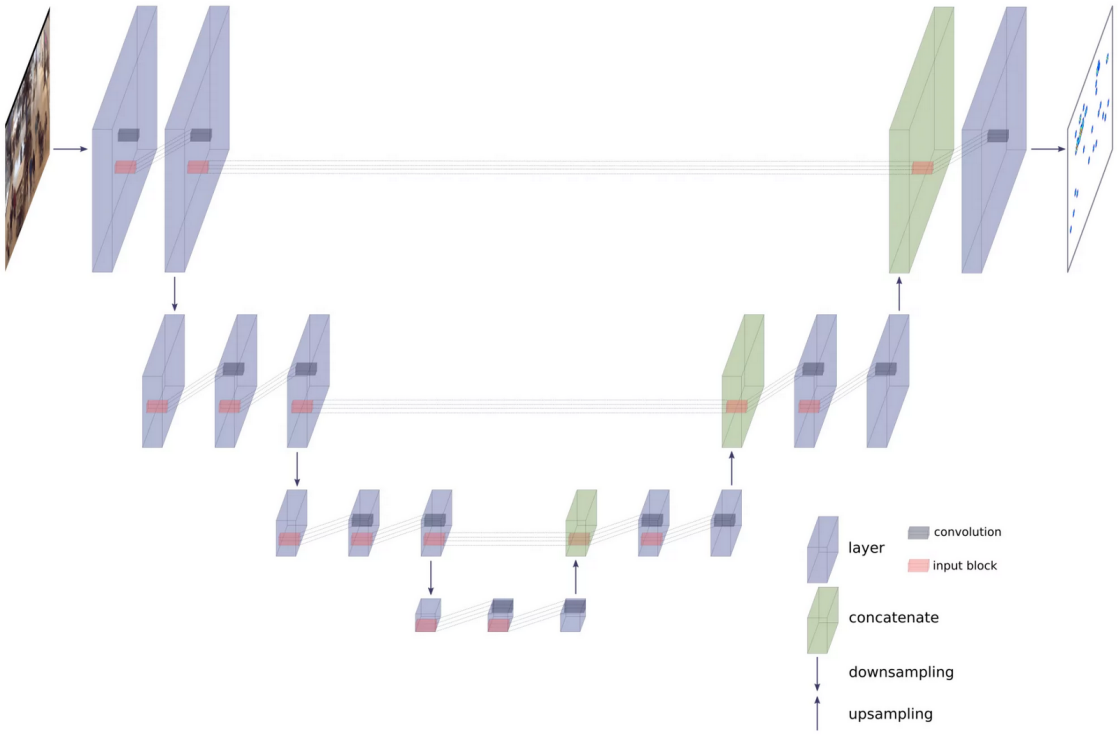

Figure 12: The U-Net architecture used, where a density map is produced at the end (far-right). [5]

5.4 Variables

5.4.1 Control

The control parameters pertaining to the model are listed in Table 1 for replicability. For all intents and purposes, a complete understanding of these is beyond the scope of this paper.

Variable	Value
Learning rate	0.01
Batch size	8
U-Net filters	64
Convolution layers per block	2

Table 1: The control parameters used for training and validation of the model.

5.4.2 Independent

To effectively answer the RQ, this paper seeks to vary the proportion of synthetic data used for training. More specifically, five qualitative variations in the training sets are examined for each dataset, as recorded in Table 2.

Name	Images	
	Real	Synthetic
Real	N	0
Synthetic	0	N
A	N	N
B	N	3N
C	N	5N

Table 2: Five variations in training dataset composition, where N is the size of the original, natural dataset.

5.4.3 Dependent

Three dependent variables were used to compare the performances of models trained on different training data. The first and foremost of these is MAE and was explained in 1. To calculate precision and recall, as defined in equations P and R respectively, the following steps were performed.

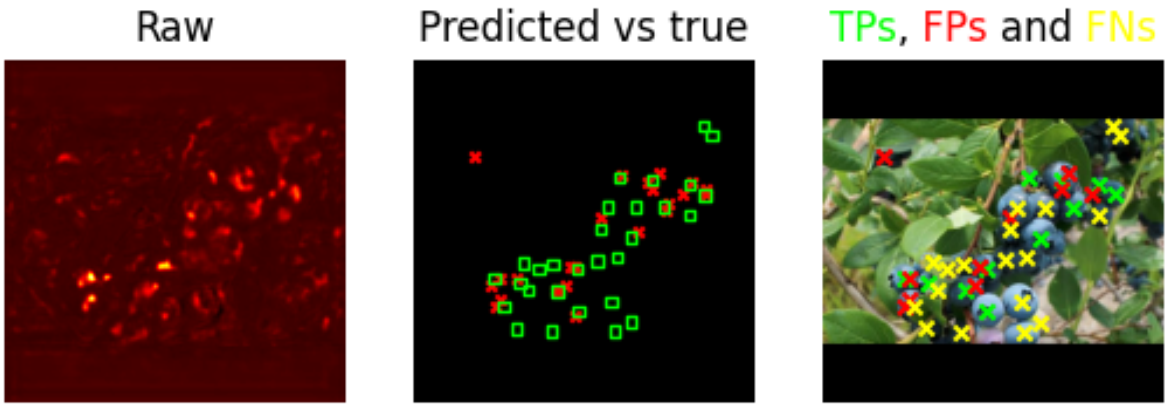

Figure 13: The visualization of the procedure used to determine precision and recall values, starting from a raw output image and using the ground-truth annotation.

First, for each training input, the raw output density map was integrated to find the predicted count of objects, N. Next, the locations of N most prominent[4] objects (peaks) in the density map were extracted, as represented by red crosses in the center of Figure 13. Each coordinate was then cross-checked against the ground-truth density map and was considered a true positive if it was positioned within a 9x9[5] pixel region of the true value, represented by green squares. Otherwise, it was tallied as a false positive. An

[4]Prominence here is defined as the brightness of a detected feature on the density map, i.e. the model's confidence in it being a TP.

[5]These dimensions were chosen because the annotation dots (see Figure 6 on page 10) were blurred by a 9x9px kernel to produce a ground-truth density map, though it is still somewhat arbitrary.

inverse process was repeated to find the false negatives: the model was considered to have failed to locate a feature if there was no predicted peak within a 9x9 pixel range of the ground-truth coordinates. The classified predictions are presented in the right-most image of Figure 13 in their respective colors: green for TP, red for FP, yellow for FN. This procedure was repeated for all images in the set and the model's precision and recall were taken to be the arithmetic mean of the computed values. The F1-score was then obtained from these two metrics, as defined in equation F1.

5.5 Data augmentation

Standard data augmentation was performed during training on images of (only) the training set. More specifically, each input image had a 50% chance of being mirrored horizontally, a 50% chance of being mirrored vertically, and a 50% chance of being randomly rotated by either 90, 180 or 270 degrees.

5.6 Training

The training was performed in a Google Colab environment with access to cloud computing resources. In particular, the T4 GPU runtime was used [6]. For each type of training set, a U-Net model was trained for 50 epochs. An epoch is defined as a single iteration during which the neural network views all images in the respective set. This number was chosen as it was noticed that the improvements in performance of models generally levelled off past that threshold (see Figure 14).

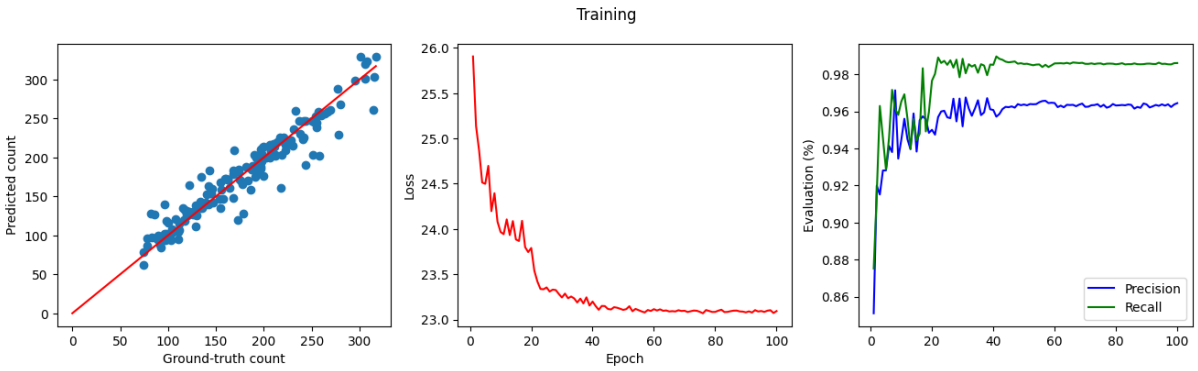

Figure 14: A single iteration of training on purely real training data from the 'cells' dataset. A similar stagnation of loss was observed for the other two datasets.

5.7 Validation

After *each* training epoch, the five differently trained networks were validated against identical data taken from the real dataset. At the end of 50 epochs, the precision and recall metrics from the epoch with the best MAE were chosen to represent the model's performance. In other words, MAE was used as the main metric due to it being the most representative of the task at hand (counting objects). Conversely, precision and recall were included to provide valuable insight into the model's performance in localization.

An example of the validation process corresponding to the training plot introduced above is visualized in Figure 15. To obtain final experimental data, the entire procedure of training and validation for 50 epochs was repeated three times.

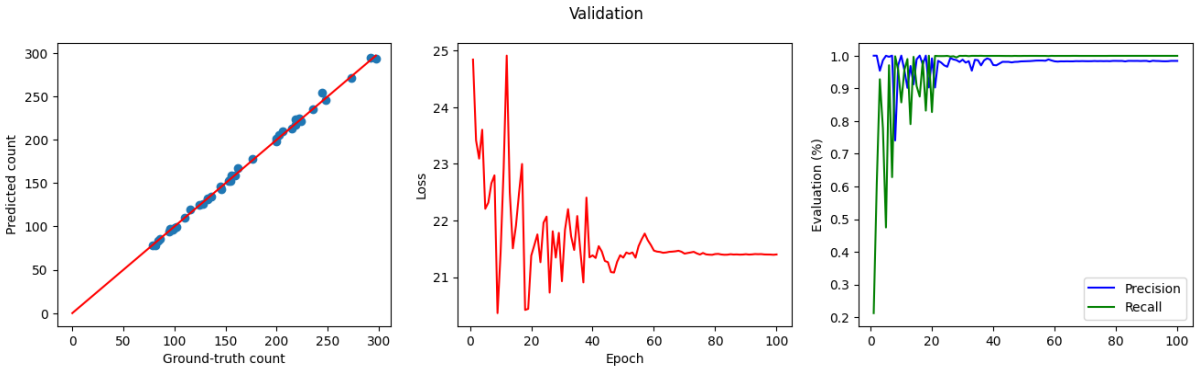

Figure 15: A single iteration of validation carried out in parallel with the training displayed in Figure 14.

6 Experimental results

For simplicity's sake, the MAE and F1-score scores presented in Table 3 are in the form of the readily computed arithmetic mean and uncertainty obtained from the three trials performed. The raw data and its processing is included in Appendix B.

Dataset	Name	MAE[↓]	F1(%)[↑]
Cells	Real	**2.24(±0.29)**	**99.1(±0.9)**
	A	6.49(±1.91)	98.7(±4.3)
	B	7.46(±5.11)	99.1(±2.2)
	C	8.21(±3.54)	97.4(±3.8)
	Synthetic	8.49(±2.95)	96.9(±4.3)
Blueberries	Real	8.78(±0.78)	21.6(±12.2)
	A	**7.19(±1.17)**	44.7(±8.3)
	B	8.43(±0.79)	44.3(±9.5)
	C	11.53(±0.43)	**55.3(±11.6)**
	Synthetic	12.07(±0.22)	24.8(±13.1)
Tickets	Real	**9.63(±0.63)**	16.7(±20.8)
	A	10.14(±0.47)	**18.1(±6.0)**
	B	11.64(±1.73)	10.6(±6.5)
	C	10.43(±1.23)	11.1(±8.3)
	Synthetic	10.02(±0.71)	11.5(±11.6)

Table 3: Computed MAE and F1 scores and their uncertainties from the three trials performed for each data point, as named in section **??**. The arrows in the headers represent whether a smaller or larger quantity is better. **Bolded** values represent the best score for that dataset.

7 Discussion

7.0.1 Cells

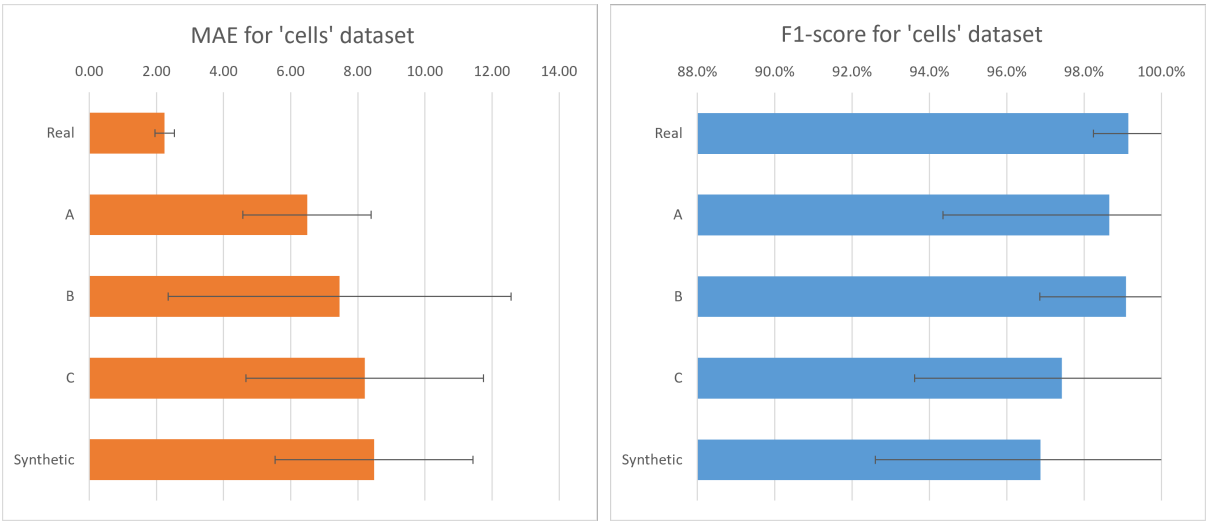

Figure 16: The MAE and F1 scores for differently trained models of the 'cells' dataset with error bars to represent uncertainty.

As expected, provided an equal number of images, the model trained on real data greatly outperformed that trained on synthetic data in terms of both MAE and F1-score, as shown in Figure 16. However, contrary to the hypothesis, bolstering the total number of images with synthetic equivalents did not benefit the model. In fact, as the proportion

of artificial data increased, the neural network's performance declined according to both MAE and the F1-score. The most likely explanation for this is overfitting, meaning that the model grew overly specialized on the synthetically generated images. Since these do not perfectly match the real validation images, the neural network simultaneously became less proficient for natural data. This is supported by Figure 17 which shows that the neural network trained on 'synthetic' data optimized its loss for the (synthetic) training data, but that it did not translate into better performance for the (natural) validation set.

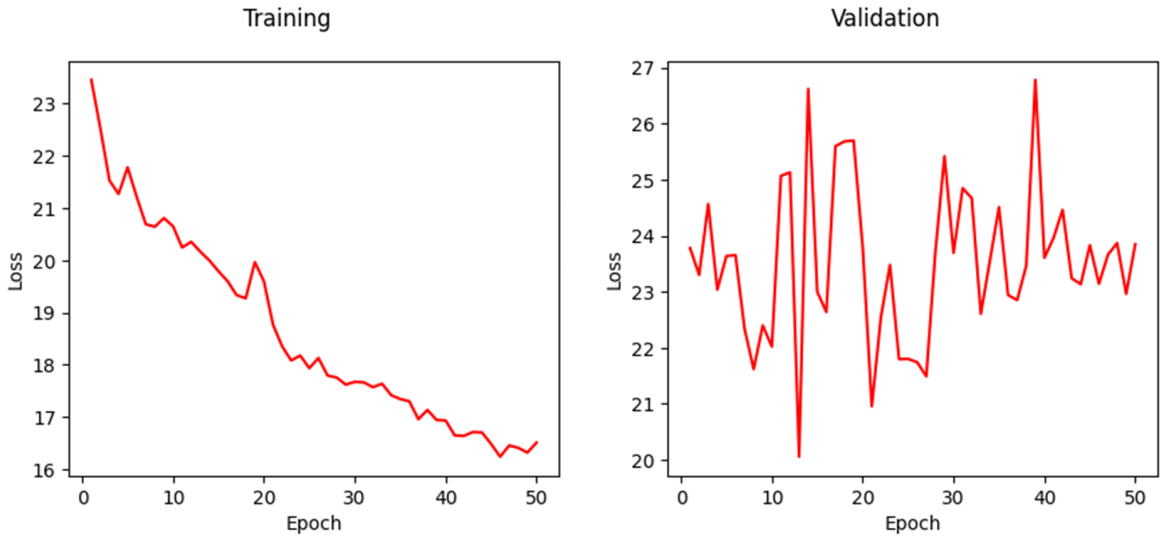

Figure 17: The discrepancy between the training and validation losses of a model trained on purely synthetic 'blueberry' data and validated against its real counterpart.

Overall, the reduction in performance is a curious finding considering that the 'VGG cells' dataset is itself artificial. Theoretically, this implies that both 'real' and 'synthetic' variations of training data employed in the experimentation (refer to Figure 16) should produce similar results. However, this discrepancy can likely be explained by the differences in how data was synthesized in this study and in [1]: the latter is much more sophisticated and representative of reality than the naive technique of 'copying and pasting' used in this paper. Going back to the RQ, the findings from this dataset suggest that in a case where a substantial amount of accurate data is readily available and is not overly complex, synthetic data (unless of a very high standard) may not be a viable solution to increase a model's performance: neither for object counting nor localization.

7.0.2 Blueberries

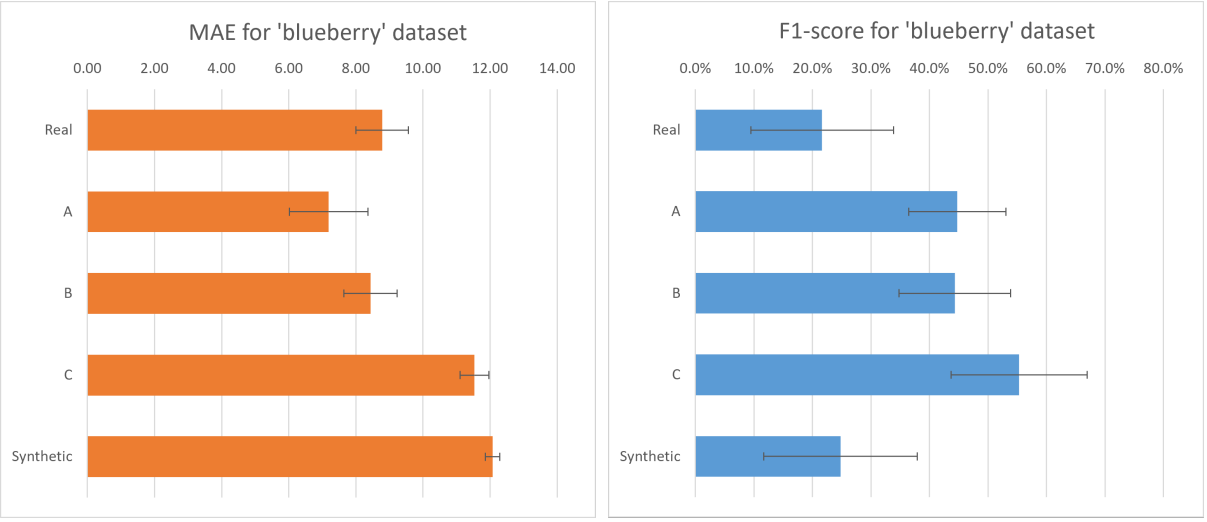

Figure 18: The MAE and F1 scores for differently trained models of the 'blueberries' dataset with error bars to represent uncertainty.

For the 'blueberries' dataset, the neural networks trained on 'A' and 'B' sets outperformed that trained on purely real data in terms of both MAE and the F1-score, as seen in Figure 18. Nevertheless, the model's performance for object counting declined with the proportion of synthetic data, just as observed with the 'cells' dataset, likely for a similar reason of overfitting. On the contrary, interestingly, the F1-score only progressed with the size of training data, irrespective of how much synthetic data there was. More investigation would have to be performed to determine if and at what concentration would the F1-score decline.

On the whole, the findings show that for a relatively complex dataset that only contains 78 readily available real images, synthetic data can be a viable method of improving the model both for object counting and localization, provided that real data isn't overly diluted by it.

Nevertheless, as mentioned, the synthesis of images in this paper is not only simplistic, but also heavily correlated with original data, since that's where the object samples are cut out from. This means that any biases present in real data can permeate through. For instance, Figure 19 shows the output produced by the model trained on set 'A'. While the neural network outperformed its counterpart trained on the 'real' set, it regardless fails to recognize unripe blueberries because those were under-represented across all training data. An extension upon this investigation could look into ways of generating artificial

data that are also able to account for biases present in natural data.

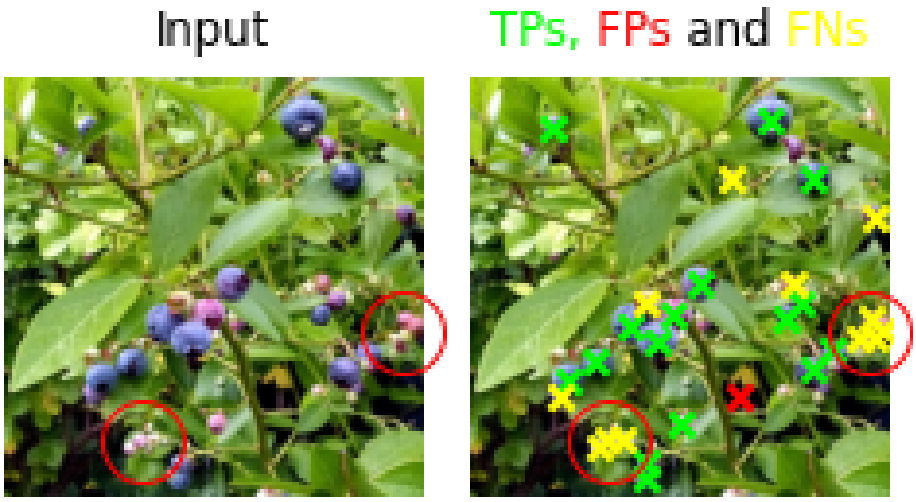

Figure 19: The failure of the model trained on training set ‘A’ to identify unripe blueberries (circled red).

7.0.3 Tickets

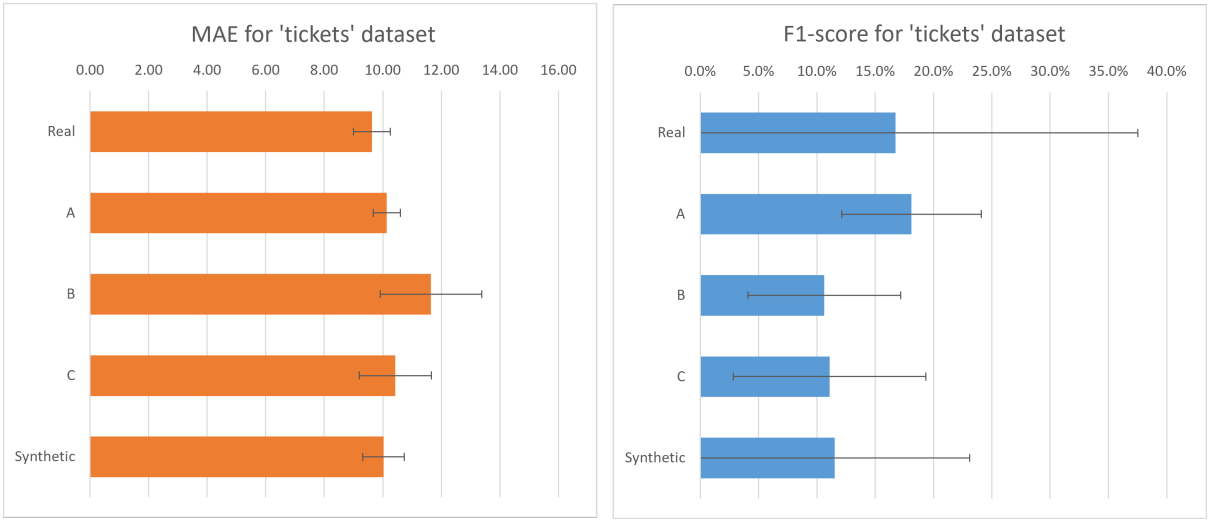

Figure 20: The MAE and F1 scores for differently trained models of the ’tickets’ dataset with error bars to represent uncertainty.

Finally, for the ‘tickets’ dataset, the neural network with the lowest MAE was trained on the ‘real’ set, while the model with the highest F1-score consumed training set ‘A’. Overall, the results are closely tied, which is likely a reflection of the complexity of recognizing extremely occluded tickets. In other words, irrespective of the training data, it was observed that models struggle to improve their accuracy past a certain threshold, as shown by a sharp initial drop in loss and subsequent plateaus in Figure 21. In this

case, improvements would likely require adjusting the model or reimagining the structure of training data, e.g. by using line segments to annotate tickets instead of simplistic dots. From a practical perspective, a synthetic approach would offer a unique advantage in this, as it would require significantly less effort to reconstruct a dataset than to undergo manual data annotation again.

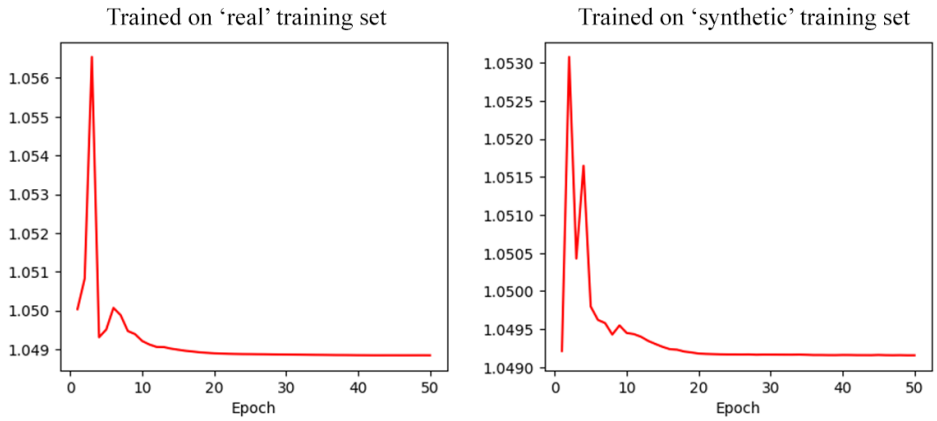

Figure 21: The loss determined from validation on identical, real 'tickets' data of two differently trained models.

8 Conclusion

Overall, this investigation sought out the viability of using synthetically generated images to train a computer vision model for real-world scenarios, specifically in terms of object counting and localization. For this purpose, three datasets of different properties were examined. The findings from these converge on the fact that artificial data is a viable method of improving the performance of a density-estimation based model, provided that the initial dataset is scarce, and the generated data closely corresponds to the one used for validation.

These results should be considered in context, as there are numerous limitations associated with this paper. First and foremost, as mentioned, the simplistic path to synthesizing data in this investigation fails to capture all kinds of artificial data. For instance, more complete techniques could involve accounting for biases in the original sets, or even creating an entirely separate neural network tasked with synthesizing data.

Secondly, the experiments were performed on a specific architecture of the U-Net model. This means that results may vary based on the type of network, as well as its

implementation. Furthermore, the parameters pertaining to the training procedure (e.g. number of epochs, learning rate and data augmentation) were left unchanged across all datasets. Nevertheless, these may have a quantifiable impact on the performance of the differently trained models. For instance, varying the transformations employed as part of data augmentation for the synthetic training set could alleviate the harms of overfitting.

Finally, while the datasets were chosen in an attempt to cover a broad spectrum, the results are still inherent to them and can not necessarily be extrapolated to other data. For instance, the conclusions obtained from the 'VGG cells' dataset are not necessarily representative of the viability of using synthetic data for datasets containing cells more generally. Furthermore, the exact boundaries of situations when artificial data benefits a training set, and in which proportion to the real data, could be accounted for in an extension to this study.

8.1 Loop closure

Figure 22: The predictions of the model overlaid on a panorama image of the wall of tickets.

Lastly, the idea for this investigation arose from a question of curiosity, so it is only fitting to conclude by answering it. In the end, the total number of tickets determined to be on the wall by the best-performing model is roughly $\mathbf{3844 \pm 1472}$, as shown in Figure 22. The specifics of the procedure are addressed in Appendix C.

References

[1] A. Lehmussola et al. "Computational Framework for Simulating Fluorescence Microscope Images With Cell Populations". In: *IEEE Transactions on Medical Imaging* 26 (July 2007), pp. 1010–1016. DOI: 10.1109/tmi.2007.896925. URL: https://ieeexplore.ieee.org/document/4265752 (visited on 11/17/2022).

[2] Victor Lempitsky and Andrew Zisserman. *Learning To Count Objects in Images*. 2010. URL: https://papers.nips.cc/paper/2010/file/fe73f687e5bc5280214e0486b273a5f9-Paper.pdf (visited on 08/22/2023).

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.org, May 2015. URL: https://arxiv.org/abs/1505.04597.

[4] Megan Beck Libert and Barry. *The Machine Learning Race Is Really a Data Race*. MIT Sloan Management Review, Dec. 2018. URL: https://sloanreview.mit.edu/article/the-machine-learning-race-is-really-a-data-race/ (visited on 09/28/2023).

[5] Tomasz Bonus and Tomasz Golan. *Objects Counting by Estimating a Density Map With CNNs*. NeuroSYS, July 2019. URL: https://neurosys.com/blog/objects-counting-by-estimating-a-density-map (visited on 08/25/2023).

[6] Google Colab. *Google Colaboratory*. colab.research.google.com, Nov. 2019. URL: https://colab.research.google.com/github/d2l-ai/d2l-tvm-colab/blob/master/chapter_gpu_schedules/arch.ipynb (visited on 09/29/2023).

[7] Wikipedia Contributors. *Precision and recall*. Wikipedia, Apr. 2019. URL: https://en.wikipedia.org/wiki/Precision_and_recall (visited on 09/29/2023).

[8] DeepAI. *Harmonic Mean*. DeepAI, May 2019. URL: https://deepai.org/machine-learning-glossary-and-terms/harmonic-mean (visited on 09/29/2023).

[9] Yue Guo et al. "SAU-Net: A Universal Deep Network for Cell Counting". In: *ACM-BCB ... ...: the ... ACM Conference on Bioinformatics, Computational Biology and Biomedicine. ACM Conference on Bioinformatics, Computational Biology and Biomedicine* 2019 (Sept. 2019), pp. 299–306. DOI: 10.1145/3307339.3342153. URL: https://pubmed.ncbi.nlm.nih.gov/34046647/ (visited on 09/04/2023).

[10] IBM. *Computer Vision*. IBM, 2019. URL: https://www.ibm.com/topics/computer-vision (visited on 08/24/2023).

[11] Di Kang, Zheng Ma, and Antoni B. Chan. "Beyond Counting: Comparisons of Density Maps for Crowd Analysis Tasks—Counting, Detection, and Tracking". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29 (May 2019), pp. 1408–1422. DOI: 10.1109/TCSVT.2018.2837153. URL: https://ieeexplore.ieee.org/abstract/document/8360001 (visited on 04/06/2022).

[12] Sofiane Sahir. *Canny Edge Detection Step by Step in Python—Computer Vision*. Towards Data Science, Jan. 2019. URL: https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123 (visited on 09/28/2023).

[13] Gerard Andrews. *What is synthetic data?* The Official NVIDIA Blog, June 2021. URL: https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/ (visited on 09/28/2023).

[14] Aysegul Takimoglu. *What is Data Augmentation? Techniques, Benefit and Examples*. research.aimultiple.com, Apr. 2021. URL: https://research.aimultiple.com/data-augmentation/ (visited on 09/29/2023).

[15] Grace Kim, Will Song, and Lucia Zheng. *Deep Learning for Cell Segmentation*. 2022. URL: http://cs231n.stanford.edu/reports/2022/pdfs/144.pdf (visited on 08/25/2023).

[16] Shijie Li, Thomas Ach, and Guido Gerig. *Improved Counting and Localization from Density Maps for Object Detection in 2D and 3D Microscopy Imaging*. 2022. URL: https://arxiv.org/pdf/2203.15691.pdf (visited on 09/04/2023).

[17] Inkyu Sa et al. "deepNIR: Datasets for generating synthetic NIR images and improved fruit detection system using deep learning techniques". In: *Sensors* 22 (June 2022), p. 4721. DOI: 10.3390/s22134721. URL: https://arxiv.org/abs/2203.09091 (visited on 08/22/2023).

[18] Adam Zewe. *In machine learning, synthetic data can offer real performance improvements*. MIT News — Massachusetts Institute of Technology, Nov. 2022. URL: https://news.mit.edu/2022/synthetic-data-ai-improvements-1103 (visited on 09/28/2023).

[19] IBM. *What is synthetic data? — IBM*. www.ibm.com, 2023. URL: https://www.ibm.com/topics/synthetic-data (visited on 09/04/2023).

[20] Paul Ewuzie and Yonas Chanie. *1-785 Introduction to Deep Learning -Spring 2023 - Recitation 8: RNN Basics Carnegie Mellon University*. URL: https://deeplearning.cs.cmu.edu/S23/document/recitation/Recitation8/IDL_S23_Recitation_8__RNN_Basics.pdf (visited on 09/29/2023).