applied sciences

MDPI

Article

Improving Robot Perception Skills Using a Fast Image-Labelling Method with Minimal Human Intervention

Carlos Ricolfe-Viala * and Carlos Blanes

Automatic Control and Industrial Computing Institute, Universitat Politécnica de Valencia, 46022 Valencia, Spain; carblac1@ai2.upv.es

* Correspondence: cricolfe@upv.es

Featured Application: Natural interface to enhance human–robot interactions. The aim is to improve robot perception skills.

Abstract: Robot perception skills contribute to natural interfaces that enhance human–robot interactions. This can be notably improved by using convolutional neural networks. To train a convolutional neural network, the labelling process is the crucial first stage, in which image objects are marked with rectangles or masks. There are many image-labelling tools, but all require human interaction to achieve good results. Manual image labelling with rectangles or masks is labor-intensive and unappealing work, which can take months to complete, making the labelling task tedious and lengthy. This paper proposes a fast method to create labelled images with minimal human intervention, which is tested with a robot perception task. Images of objects taken with specific backgrounds are quickly and accurately labelled with rectangles or masks. In a second step, detected objects can be synthesized with different backgrounds to improve the training capabilities of the image set. Experimental results show the effectiveness of this method with an example of human–robot interaction using hand fingers. This labelling method generates a database to train convolutional networks to detect hand fingers easily with minimal labelling work. This labelling method can be applied to new image sets or used to add new samples to existing labelled image sets of any application. This proposed method improves the labelling process noticeably and reduces the time required to start the training process of a convolutional neural network model.

Keywords: human–robot interactions; image labelling; deep learning; image classification

Citation: Ricolfe-Viala, C.; Blanes, C. Improving Robot Perception Skills Using a Fast Image-Labelling Method with Minimal Human Intervention. *Appl. Sci.* **2022**, *12*, 1557. https://doi.org/10.3390/app12031557

Academic Editors: Juan Jesús Roldán-Gómez and Mario Andrei Garzón Oviedo

Received: 23 December 2021
Accepted: 25 January 2022
Published: 31 January 2022

1. Introduction

Robot perception skills have been significantly improved by the use of deep neural networks in image classification, as well as object detection and segmentation [1–6]. As it is the closest to human vision, segmentation is the most powerful of these and can be applied to a wide range of tasks. Although these advancements are satisfying, they also present many new challenges [7,8].

Object detection and image segmentation tasks use supervised learning methods that require huge quantities of correctly labelled data to feed the training process. For object detection, a rectangle defines the position of the object in the image. For object segmentation, a set of pixels determines the object silhouette with a mask. To obtain this amount of data, hundreds of hours of monotonous manual work is usually needed. For example, the ImageNet challenge [9] has more than one million images classified into one thousand different categories. Each image has to be labelled into its corresponding category by a human. Moreover, if the classification of image objects is categorized semantically, the effort required to label the objects exponentially increases and risks of bad labelling arise.

Object labelling tools include image-sharing options that help offset the tediousness of the manual labelling process. This may enable the outsourcing of the labelling task,

with labellers motivated by compensation or personal interest in the subject. Furthermore, image sharing may aid the regulation of the labelling process, allowing further checks of the accuracy of masks and the consistency of labels. It is very important to establish a data quality control system to edit and review object images that have been skipped or badly labelled. Online tools working from browsers help with this task. With an online tool, images can be easily uploaded, and it is therefore easy for a team of people to work on a labelling task. Offline tools require installation before use, and if conducting a team labelling task, they also need cloud storage. Some online and offline labelling tools include multi-user access as a feature for team and project management. This allows co-working between an internal team of experts or managing staff that label the data. Other tools allow the option to automatically track the staff consensus on labels. Many tools also offer labelling metrics that monitor the time taken to label each object and the labelling activity per labeller. This information helps to ascertain labelling costs.

To notably reduce the effort required to label the object, some manual object labelling tools propose an objects region with algorithms that infer the boundary of the object to separate the background and the foreground. They use 2D computer vision algorithms for object detection that perform a silhouette analysis, which could help to identify object pixels in the image [10]. However, as obtaining an object silhouette is not an easy task, this method can only be applied in specific situations. Object colour features change depending on where in the image the object is due to illumination. To perform a successful object segmentation that creates a silhouette, it is necessary to control illumination and to have a uniform background. In addition, object colour influences the silhouette detection process as pixel intensity plays an important role in the threshold process. To resolve this problem, some approaches convert images from RGB to HSV colour spaces where object colour is easier to define. Furthermore, with an object contour or silhouette as a region of interest (ROI), it is necessary to extract some scale and time invariant features to decide if it represents an object of interest or not. For example, to detect objects in images, convex defect detection measures the ratios between convex hull areas and object silhouette areas [11].

To resolve the problems with illumination, object colour changes and controlled backgrounds, techniques that focus on detection of edges have arisen. The starting point for these techniques is the gradient of image intensity, which increases its robustness against changes in lighting, object colour and uncontrolled backgrounds. The gradient of the image highlights the edges in the image. Consequently, an edge analysis allows the extraction of hard features that are dependent on the object shape and independent of the colour of the pixels. For example, Chaudhary [12] used the histogram of oriented gradients technique to extract features and classify the gestures of bare hands with different skin colours and illumination. The orientation histogram is a technique developed by Liversidge [13] and improved by Dalal and Triggs [14] in their work, which focused on human detection in images and videos. Another technique that may help to detect objects in sequential frames of video footage is the Kanade–Lucas–Tomasi algorithm [15]. These techniques can help with the object labelling process but they are not able to perform an accurate labelling process independently. With 2D computer vision algorithms, the accuracy is not as good as required in images with generic backgrounds, and the approach to object pixels proposed above is the starting point for manual labelling. Changes in backgrounds, foregrounds and object occlusion in images make the automatic and reliable detection of objects in generic images a very complex problem.

Labelling with transfer learning uses neural network models for initial annotations of label-specific elements in an image. Some tools integrate third-party APIs, such as ImgLab, which is integrated with face++ API; faces can therefore be labelled as faces by marking the significant aspects of the image. Additionally, VIA [16,17] offers face bounding box tracking with Faster RCNN [18]. Matlab Image Labeler App includes a built-in automation algorithm to detect and label people and vehicles using a pre-trained detector based on aggregate channel features. The Ground Truth Labeler of Automated

driving tool implements the Kanade–Lucas–Tomasi [19–21] algorithm to track features in successive frames of a video, labelling in bounding boxes only. Vicomtech [22] and Mask Editor [23] classify object parts into super-pixels, grouping nested pixels of a similar colour and considering that the borders of object parts have strong colour gradients. Afterwards, a group of super-pixels defining object parts are labelled manually as a complete object. If no shadows appear in the image, the accuracy in the borders with irregular shapes is considered better than a human-setting polygon vertex.

Unfortunately, even after fine-tuning the parameters, the performance of automatic detection algorithms is disappointing due to the great variation in the perspectives of objects, as well as image conditions, backgrounds and foregrounds. Moreover, this strategy fails if a pre-trained model does not exist or if the pre-trained model does not classify the new image set with a good success rate. However, this transfer learning method could be a good starting point from which to manually drag and adjust the suggested area, therefore improving label accuracy. New images classified with a pre-trained model are the starting point of manual labelling, consisting of revising all work completed by the transfer learning process and fixing bad annotations if necessary.

In addition, several authors use image synthesis to reduce the effort of manual annotations. Gupta et al. [24] localized text in natural images, synthetizing computer-generated texts and natural real images. Su et al. [25] and Sun and Saenko [26] synthetized 3D CAD object models with real background images. Castro et al. [27] generated synthetic structural magnetic resonance images for learning schizophrenia. Segawa et al. [28] recognized first-person reading activity by synthetizing computer-generated images and real background images.

The aim of this paper is to propose a novel method for fast image objects labelling with minimal human intervention to create new data sets easily that can then be used in the training process of a neural network model. The proposed method can be used with a new set of images or to improve existing image sets when introducing new samples. The proposed fast image-labelling tool defines an image background conditions in detail to help 2D vision algorithms to detect objects in images quickly and accurately. The results of model training with fast-labelled images are similar to the results of model training with manually labelled images. An analysis of the effects of image background in the model training process demonstrates that the proposed method is valid. The contribution of this paper is the reduction in the collection and annotation costs of deep learning datasets by using simple object detection and image synthesis.

2. Materials and Methods

Automatic image labelling to train deep neural network models is a very arduous computer vision task. The aim is to detect objects as group of pixels in the image, to teach a model to identify similar objects in new images. Is it possible to perform fast object labelling in images with no human intervention? The answer is yes, provided that some conditions are controlled. Regarding objects in images with controlled background, 2D computer vision algorithms are suitable for their quick detection. With a controlled background, it is possible to perform object segmentation that accurately assesses object silhouettes. It can also detect changing object colours, location or orientation in the image. Images with objects in controlled backgrounds allow for fast object detection. A controlled background is defined as a uniform colour that is different to the object colours. If the background is controlled, object silhouettes can be generated using 2D computer vision algorithms.

Currently, the proposed technique requires images similar to those in Figure 1a, in order to label new images quickly. These images have constant backgrounds and objects that are significantly different to the background. With images similar to those in Figure 1a, object segmentation is extremely easy to achieve using 2D computer vision techniques, and this segmentation allows for the automatic labelling of new images. Pixel selection is carried out by value easily, since object pixels are quite different from background pixels. The process is shown in Figure 1. Suggested object regions for labelling purposes are

definitive in 99.9% of cases and are, therefore, unlikely to need any manual modification. In a few milliseconds, an image is labelled and ready to be used for training a model.

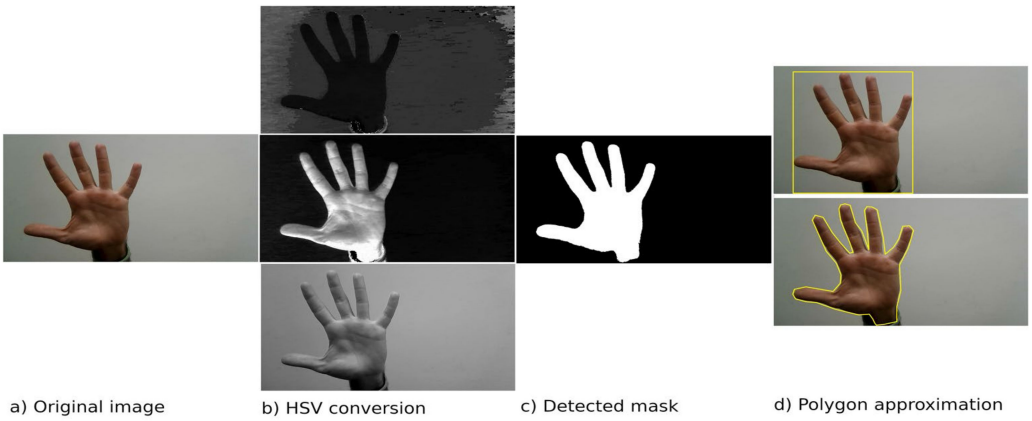

Figure 1. Fast image-labelling process. Original RGB image (**a**) is converted to HSV colour space to increase differences between background and object pixels (**b**). Colour saturation layer is very useful for selecting object pixels and creates a mask (**c**). Detected mask is adjusted to a rectangle or a polygon if necessary (**d**). Outcome of the automatic image labelling process is a mask saved as a png image, or a region of interest defined with a rectangle in an xml file or a polygon in a json format.

Moreover, it is possible to perform data augmentation and create new images easily, as shown in Figure 2. Using 2D computer vision techniques, the proposed algorithm for fast image-labelling is illustrated in Figures 1 and 2 as follows:

1. Define a controlled background with a constant colour. In addition, the background colour has to be different from the colours of the objects. An example is shown in Figure 1a.
2. Produce images with a controlled background. The viewfinder should be framed by the controlled background and the objects clearly inside the viewfinder. If part of the object is outside of the viewfinder, it can be considered an occluded object.
3. If necessary, exacerbate the differences between the object and background pixels. As discussed, the aim is to obtain an image where background pixel colour is easy to differentiate from the object pixels. If it is necessary to accentuate this difference, two operations are proposed:
    a. If necessary, perform an RGB to HSV conversion to highlight the object pixels in the image. Some objects such as white skin tones are easier to detect if they are in a colour space different from RGB. This operation is represented in Figure 1.
    b. Image subtraction of the background without objects from the image with objects is represented in Figure 2 as f_1. In this case, an image of the background without objects is necessary. This step will help to remove shadows and brightness in the image background and will increase the difference between object pixels and background pixels.
4. Remove the background pixels by carrying out pixel segmentation. With an image similar to the one shown in Figure 1b, the contrast between object pixels and background pixels increases significantly and the detection of object pixels is easy. Pixel selection is easily carried out by value, since object pixels are quite different from background pixels. The result is a binary image where pixels that belong to object are set to 1 and pixels that belong to background are set to 0.
5. Perform a combination of dilation and erosion algorithms to close holes and noise in the segmented image.
6. Detect objects in images by grouping selected pixels in object masks.

7. Perform data augmentation, overlapping objects on images that contain only the background. This step is represented with function f_3 in Figure 2. Several objects can be synthetized with the same background to perform data augmentation.
8. Save the masks defined by the silhouette.
    a. A bounding box of the mask defines a rectangle. Rectangles are saved in an xml file.
    b. Perform polygon approximation to obtain the polygon vertex of an object silhouette to be saved in json format.
    c. Masks are saved in a png image.

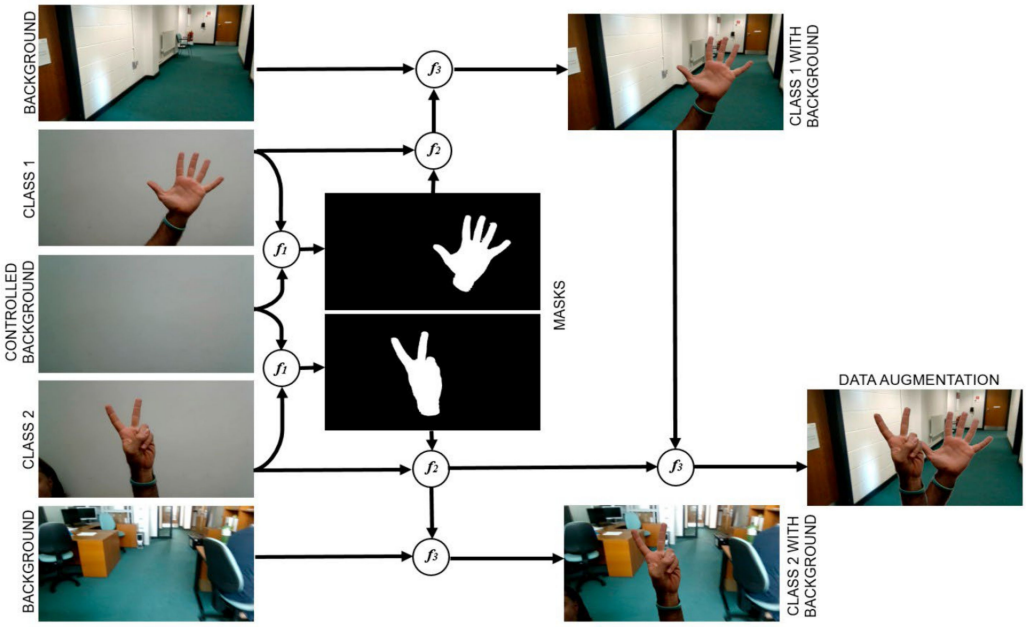

Figure 2. Proposed fast object detection algorithm with data augmentation. Images with controlled background are labelled as is shown in Figure 1. Detected objects are overlapped with a texturized background to create a standard image taken under uncontrolled conditions. Moreover, several objects can be overlapped with the same background: f_1 represents an image subtraction to obtain the object mask and f_2 is a pixel selection from original image using a mask detected with f_1. With f_3 selected pixels are synthesized with real background images to improve the training capabilities of the image set.

With this algorithm, image labelling is completed quickly and the results are saved in several formats that can immediately be used in the training process. This technique is suitable for projects that need to resolve a new problem without previous images and work. For example, many scientific or industrial quality and process control applications, need computer vision systems to make decisions. Additionally, chemical and biological researching applications need to detect elements in a microscope image to define the results of their experiments. To train these systems, a deep neural network model could be a means of solving existing problems that remain unresolved by 2D computer vision algorithms, such as the semantic segmentation of objects, even with adjoining or overlapping objects or the recognition of heterogeneous textures (e.g., plants). In these applications, it is easy to generate a set of images with controlled backgrounds.

Moreover, if it is necessary to add new images to an existing labelled image set, this controlled background technique dramatically reduces the labelling time. In both cases, to capture new images, it must be considered that controlled backgrounds are conducive to accurate object labelling. This will reduce the object labelling time noticeably and ensures accuracy of the labelling process.

This algorithm is tested in GPU Nvidia Geforce Titan XP.

3. Results

Two experiments prove the proposed method. The first experiment demonstrates the image background effects on the model training process. The second experiment tests the performance of data augmentation techniques on a new data set used to resolve a specific application.

3.1. Background Effects

The first experiment attempts to evaluate the influence of the image background in the deep model training step with a publicly available image set that offers labelled images with objects masks. This image set is 'pets' [29] and it has 200 images for each class of pet: 37 classes. The images have large variations in scale, positioning and lighting, since objects have different sizes, positions in the image and illumination. Masks are available and they allow the easy separation of cats and dogs from image backgrounds by using 2D computer vision algorithms. The aim is to train several deep learning models under different frameworks to check if the performance varies depending on the image background features. Several frameworks are tested. One framework uses cats and dogs in images with original backgrounds. Another framework takes cats and dogs alone, in images with a flat background, in the same position and orientation as the original image. The third framework utilizes data augmentation capabilities by creating new images with combinations of cats and dogs, including occlusions, varying positions and scale.

The experiment uses both flat and original backgrounds, as shown in the images in Figure 3. The first and second rows show original data from the pet image set. The third row shows objects on a flat background, created with a combination of images from the first and second rows. The fourth row shows the results of a data augmentation technique: several objects are in the image and occlusions exist. The fifth row is the resulting mask with occlusions.

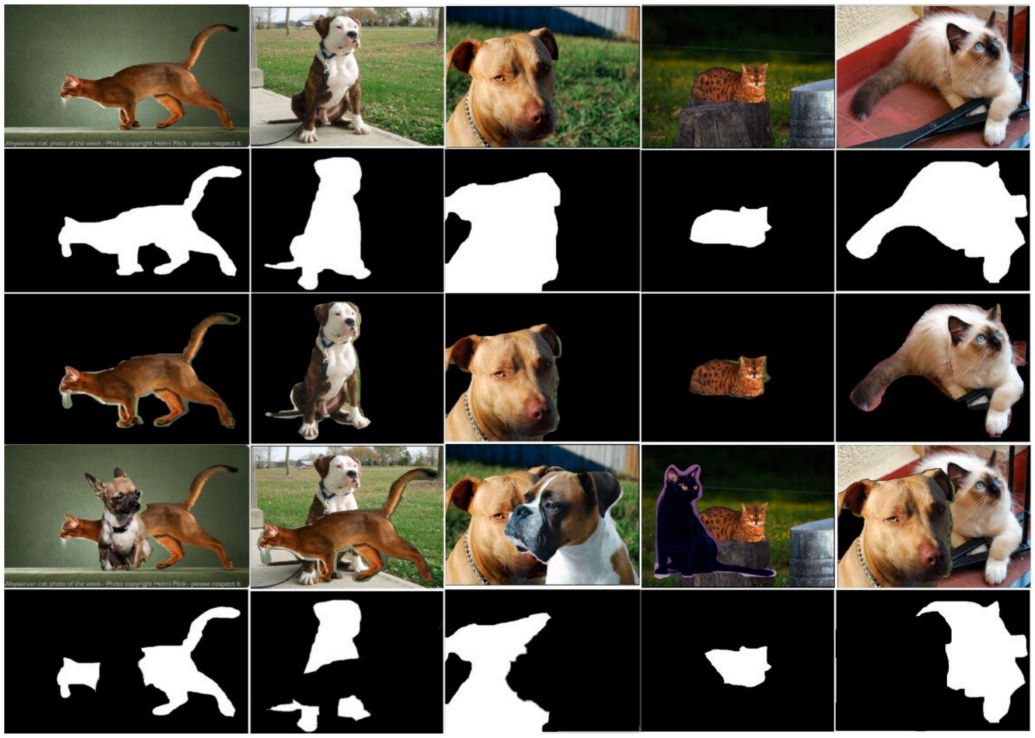

Figure 3. Image set used to test background variations. First row shows images with original backgrounds. Second row shows masks resulting from the labelling process. Third row shows images with flat backgrounds. Fourth and fifth rows show results with data augmentation techniques with resulting masks. All of these operations were carried out using 2D computer vision techniques.

Training several models allows for the comparison of their results. From this comparison, conclusions can be drawn about how image background influences the training stage of a deep neural network model. In this case, the selected models are SSD-MOBILENET, SSD-RESNET, FASTER-RCNN-RESNET and FASTER-RCNN-INCEPTION. In general, RCNN models offer better results but they are more time-consuming than SSD models. For applications where time is crucial and accuracy is not so important, it is advisable to use SSD models. Tensorflow library [30] is used and models are available in "detection model zoo" [31]. Tensorflow provides models trained to work with the COCO image set. The provided weights are the training pipeline for the pet detection model. In fine-tuning, the final layer of the model is retrained only because the fine-tuning of deeper layers degrades the performance. The parameters are optimized with the cross-entropy loss function using the stochastic gradient descent (SGD) algorithm. In the optimization, mini batches of size 10 are used. Figure 4 shows how model losses decrease with the number of epochs and training data. Variations in training time over the training data sets are not significant. Training data changes with each experiment (original images, flat background, data augmentation and data augmentation with flat background). Table 1 shows measures, such as precision and recall, of performances of the models. Models are restored at epoch 2500 and then run with training and testing data. 180 images per class are used for training, and 20 images are used for testing purposes.

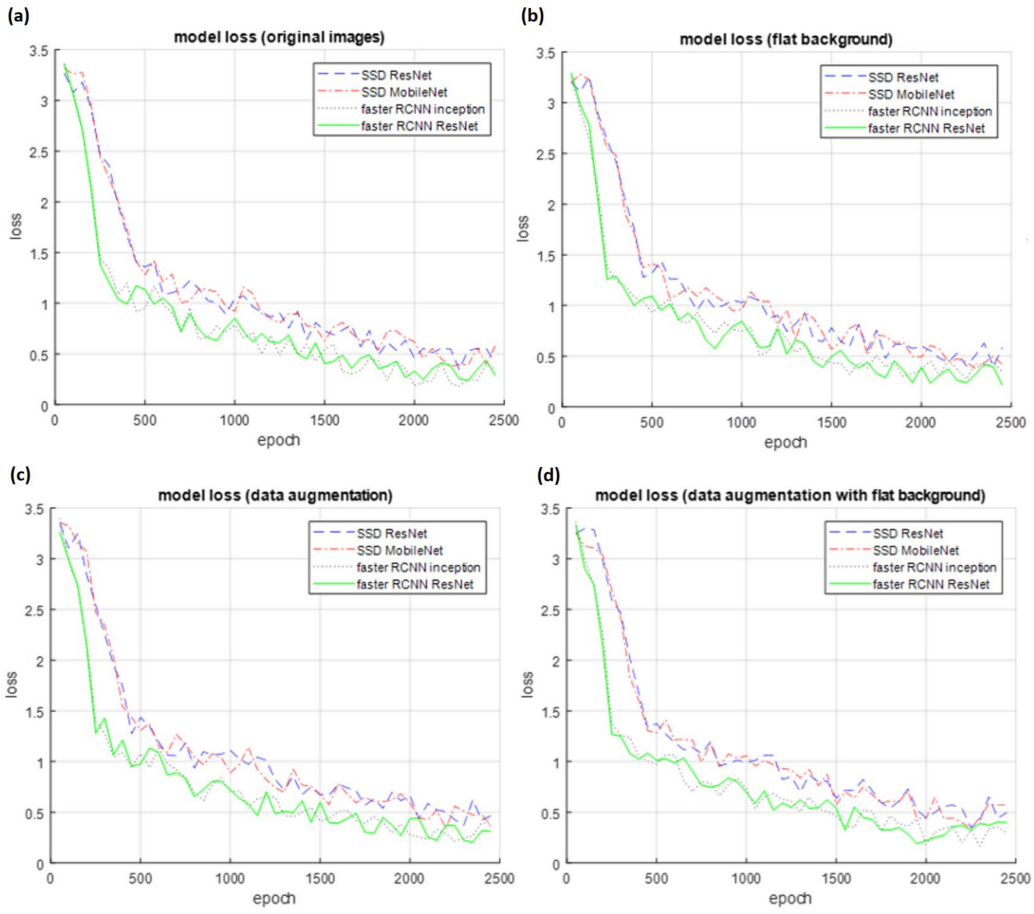

Figure 4. Models loss decreases with the number of epochs. Training data have no significant effects on the evolution of the training process.

Table 1. Detection rates (precision/recall) of deep neural network models [1] with different images data sets of pets.

		SSC		FASTER RCNN	
		MOBILENET	RESNET	RESNET	INCEPTION
training data	original images	0.81/0.88	0.80/0.74	0.85/0.81	0.84/0.86
	flat background	0.82/0.85	0.81/0.73	0.86/0.89	0.82/0.83
	data augmentation	0.77/0.87	0.78/0.75	0.80/0.89	0.83/0.84
	data augmentation flat background	0.78/0.86	0.77/0.74	0.79/0.88	0.81/0.82
testing data	original images	0.41/0.47	0.40/0.44	0.45/0.51	0.44/0.56
	flat background	0.42/0.45	0.41/0.43	0.46/0.49	0.42/0.53
	data augmentation	0.47/0.47	0.48/0.45	0.40/0.49	0.43/0.54
	data augmentation flat background	0.48/0.46	0.47/0.44	0.49/0.48	0.41/0.52

[1] Model is restored at 2500 epochs trained with different set of images. Original images are images with textured backgrounds. Flat background represents original images but with a removed background. Data augmentation are images with combinations of cats and dogs with occlusions, changing positions and scale. Data augmentation with flat backgrounds are the same images with data augmentation but with a removed background. Testing data are a set of images not used in the training step with textured background. Since the pet data set has 37 classes, it is impossible to show precision and recall for each class. Here, precision and recall are mean values of all classes. Results are similar if model training step has images with textured backgrounds or flat backgrounds.

Images with original backgrounds are always used for testing data. The first set of rows in Table 1 shows the detection rates from the training data, with different models in columns and different sets of images arranged in rows. The second set of rows of Table 1 is equal to the first rows of Table 1 but with testing data included. As the pet data set has 37 classes, it is impossible to show the precision and recall for each class. Therefore, the precision and recall are mean values of all classes.

Since the training data set is small and does not represent all pets in all positions, locations and illuminations, the detection rate when using testing data is poor. However, the differences between detection rates with training and testing images data sets depends on whether the background is textured or flat. The differences in classification with models trained with different datasets are irrelevant. This means that the effects of background in the training process are not fully representative and images with flat backgrounds are useful for training deep neural network models. In this paper, the aim is not to train a model to detect pets in images with high detection rates. The aim is to compare the rates of trained deep models and see the variations within different image data sets, in which the objects are equal but have different backgrounds. Outside the scope of this paper, to improve the detection rates of trained deep models, a representative data set of pets should be chosen. These experiments focus on evaluating how the background of data set images and data augmentation techniques can change the detection rate of a deep model.

Several conclusions arise from these results. First, pet detection is quite similar if the background is the original or a black background. Model training with the same set of objects and changing the image background does not represent significant changes in detection rates. Training a model with background images could represent an improvement of 3% or 4% in precision and recall. This fact validates the initial hypothesis that background does not significantly influence the model training process. In addition, data augmentation with a combination of objects does not considerably improve results.

3.2. Creating a New Data Set

The second experiment creates a new image set to train a deep learning model to detect hands with fingers in images. This application is very useful for enhancing human–robot interactions. Finger detection is an unresolved engineering application for which vision 2D did not find a solution. Finger detection is an extremely difficult task due to the wide variety of shapes they can display in an image. Fingers can be straight or curved, partially occluded, grasping other things, or other hands, and seen from different viewpoints. Research in this area is underway as the use of hands in robot interfaces is a very attractive method for human–computer interaction [32,33].

To train a hand detection model, a set of 20,000 images of human fingers are labelled in a few minutes using the proposed fast image-labelling algorithm. Classes are "one finger", "two fingers" and so on. Figure 5 shows one sample for each class. The first row shows images of hands with a controlled background for which the detection of the hand silhouette was easy. The second row shows masks extracted using the algorithm described in Section 3. The third row shows the resulting images of combining hands with backgrounds to improve the training process. Twenty thousand images are processed with the proposed algorithm for training and 2000 different images, similar to the image in Figure 6, are tested. As before, the aim is to show how to easily create a new data set of labelled images.

In this case, FASTER_RCNN_INCEPTION is the model chosen to perform the experiment. Table 2 shows the precision and recall of the model's performance with 2000 testing images of each class. Columns depict the hand rate detection of different subsets. Objects of the "one finger" class are classified as "one finger", but also in the "two finger" class and so on. There is similar occurrence with objects of the "two fingers" class and the successive classes. A summary of this information is in the last two rows, which gives the precision and recall of the model with each subset. Precision is tested with 400 images of each class. Recall uses the total number of images that the model classifies under each class. As can be seen, the least accurate results were those in the "three finger" class.

Regarding the background of the training images, the first set of rows in Table 2 shows the results from the testing images using a model trained with flat backgrounds, and the second set of rows in Table 2 are the results from a model trained with textured images that are made by synthetizing objects and background images. Mixing objects with textured backgrounds increases the object detection rate in 3% of the images, similar to the results from the original pet images of the previous experiment. This demonstrates that images with a controlled background are useful for the easy detection of objects and a combination of detected objects with backgrounds increases the performance of the training process.

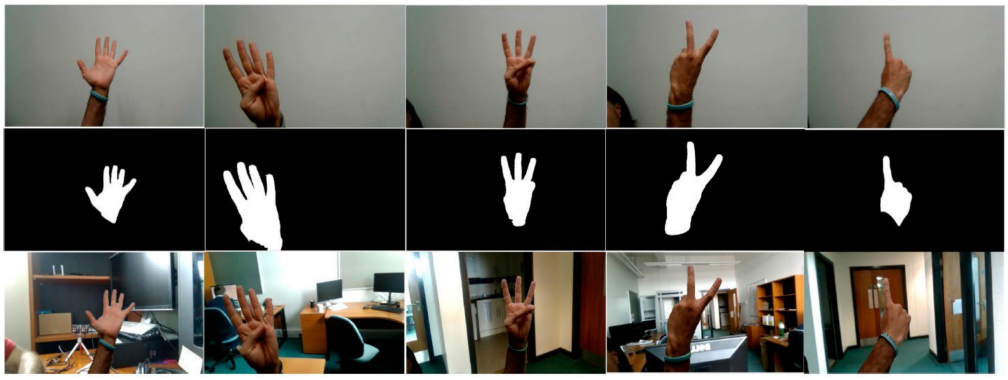

Figure 5. Labelled images using the proposed automatic image-labelling algorithm. First row shows images of hands with a controlled background to easily detect hand silhouettes. Second row shows masks extracted using algorithm described in Section 3. Third row show images where hands are combined with backgrounds to improve the training process.

Figure 6. Images to test deep model trained to detect hands with fingers.

Table 2. Detection rates (precision/recall) of deep model FASTER-RCNN-INCEPTION to detect finger hands [1].

		CLASSES				
		ONE	**TWO**	**THREE**	**FOUR**	**FIVE**
CLASSIFIED AS (%) (Training data are flat background images)	ONE	0.7000	0.1167	0.0100	0.0167	0.0033
	TWO	0.1733	0.7333	0.1600	0.0233	0.0233
	THREE	0.0767	0.1300	0.6667	0.1100	0.0300
	FOUR	0.0333	0.0167	0.1400	0.6933	0.1333
	FIVE	0.0167	0.0033	0.0233	0.1567	0.8100
	Precision	0.8268	0.6587	0.6579	0.6820	0.8020
	Recall	0.7000	0.7333	0.6667	0.6933	0.8100
CLASSIFIED AS (%) (Training data are with texturized background as combination of objects and images of backgrounds)	ONE	0.7033	0.1167	0.0100	0.0167	0.0033
	TWO	0.1767	0.7400	0.1333	0.0300	0.0167
	THREE	0.0633	0.1233	0.6833	0.1200	0.0267
	FOUR	0.0400	0.0167	0.1500	0.7000	0.1300
	FIVE	0.0167	0.0033	0.0233	0.1333	0.8233
	Precision	0.8275	0.6748	0.6721	0.6752	0.8233
	Recall	0.7033	0.7400	0.6833	0.7000	0.8233

[1] Classes are "one finger", "two fingers" and so on, which are classified as "one finger", "two fingers", etc. Precision is computed with 300 testing images of each class, similar to Figure 5. Recall is computed with the total amount of images that the model classifies under class "one finger", "two fingers", etc.

4. Discussion

Authors should discuss the results and how they can be interpreted from the perspective of previous studies and of the working hypotheses. The findings and their implications should be discussed in the broadest context possible. Future research directions may also be highlighted.

4.1. Background Effects

How do image backgrounds influence the model training process? Object labelling is crucial since it defines, with rectangles or masks, object pixels that feed the training process [2,18]. In a deep neural network model, object detection is performed in two steps. First, deep models use convolutional neural networks (CNN) that perform feature extraction based on edge detection. Second, fully connected layers, fed with edges from CNN layers, classify sets of pixels. CNN weights are adjusted to classify edges similar to the selected object edges that fed the training step. Selecting pixels through CNN layers provides features to object detection. If object masks define selected image areas, only areas with objects will train the model, and background pixels do not participate in the training process.

If a background exists, it will provide edges, but these edges are useless because they are not in the selected areas of the image provided by the object masks. Consequently, if masks define objects, background pixels do not participate in the training process because they are outside of the filter provided by the mask. Alternatively, if objects labels are rectangles, background pixels in the rectangle area will participate in the training process. However, the effect of the background pixels is not substantial because the ratio of object pixels versus background pixels is insignificant in a rectangle. In conclusion, regardless of the tool used for the identification of objects, the image background does not dramatically influence training process. The results of the conducted experiments show that the training process improves by 3 or 4% in precision and recall if images with backgrounds are used. Considering that labelling images with a background noticeably delays the beginning of the training process, working with images without a background is a valid option.

4.2. Data Augmentation

The proposed method allows for the creation of images with backgrounds. It is possible to create new images with a combination of labelled objects and standard backgrounds. Once masks define objects in images with controlled backgrounds, objects combined with images with standard backgrounds will replicate standard images taken under normal conditions for manual labelling. This process of data augmentation will easily add edges to the background.

Furthermore, several objects combined in one image will increase the training capabilities of the image data set. This combination includes occlusions and changes in positions and orientation. This technique of data augmentation allows for the easy creation of a new data set of labelled images in a few minutes. Figure 7 shows this process.

4.3. Real or Virtual Images

It could be argued that images with controlled backgrounds are closer to virtual images than real-world images. This could mean that the proposed method is not a useful method for labelling images to train a model to detect objects in real images. This is true if images from the data set only show objects from one point of view or under similar lighting conditions. A model will be able to detect objects under similar conditions to those under which the training data set was created. To improve the quality of the training data set, many perspectives of the objects in varied lighting conditions are necessary. Figure 7 shows different types of images of a hand representing the number five under different lighting conditions. Since images are intended to train a model for a specific task, this technique is extremely useful because changing the appearance, light and point of view of the objects in the image is relatively simple. However, with publicly available images, it is very difficult

to obtain a data set of this richness with multiple perspectives of the same object. Instead, it is necessary to search through thousands of images, and then to manually label all of them. With the fast labelling process described in this paper, the images in Figure 7 can be created and labelled easily and quickly.

Moreover, several authors used synthesized images to train deep learning model successfully [24–28]. This fact demonstrates that image synthetization is a technique that reduces the effort of manual annotations.

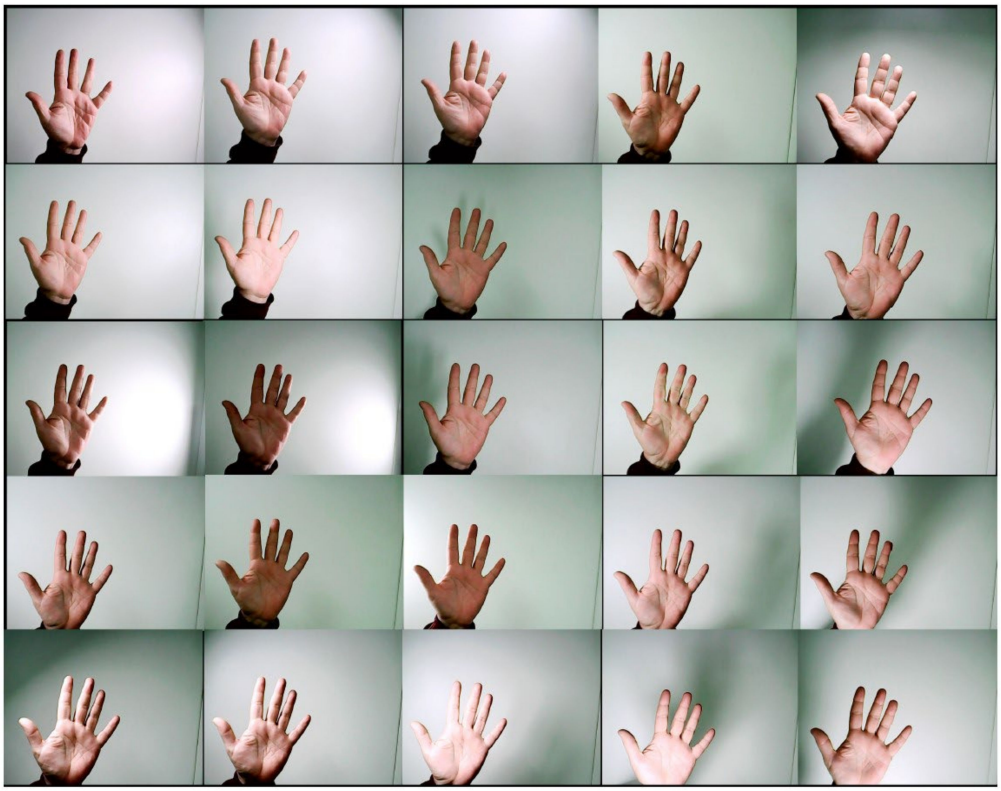

Figure 7. Different types of images of one hand representing number five under different lighting conditions. To improve the training capabilities of the training dataset, many points of view and lighting conditions of the object are necessary.

5. Conclusions

The proposed fast image-labelling algorithm represents a real alternative to manually labelling objects in images that reduce the set-up of any deep learning application. The basis of the proposed fast image labelling process is to capture images that have a controlled flat background different from the object pixels. Using 2D computer vision techniques, object detection is easy in images with this flat background with no textures. Rectangular or mask regions defining the object silhouette are computed easily.

The human–robot interaction application using fingers is treated by easily creating a newly labelled database from scratch. The detection rates of trained models using images with flat backgrounds are very similar to models trained with normal, textured backgrounds. Moreover, to improve the training capabilities of the image set, detected objects synthetized with textured backgrounds generate images similar to standard images taken under uncontrolled background conditions. In addition, data augmentation techniques such as occlusions and scaling can increase the quality of the training data set. Considering that manual object labelling is a tedious and time-consuming task, the proposed algorithm can be used to efficiently label objects in images. This algorithm is therefore a step forward in the field of image labelling that helps in any application where the training of deep learning models is a crucial step.

Author Contributions: Conceptualization, C.R.-V.; methodology, C.R.-V.; software, C.B.; validation, C.R.-V. and C.B.; formal analysis, C.R.-V. and C.B.; investigation, C.R.-V. and C.B.; resources, C.R.-V. and C.B.; data curation, C.B.; writing—original draft preparation, C.R.-V.; writing—review and editing, C.R.-V. and C.B.; visualization, C.R.-V.; supervision, C.R.-V.; project administration, C.R.-V.; funding acquisition, C.R.-V. All authors have read and agreed to the published version of the manuscript.

Funding: The Universitat Politècnica de Valencia has financed the open access fees of this paper with the project number 20200676 (Microinspección de superficies).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The source code of the proposed algorithm for fast image labeling is available at: https://github.com/cricolfe/Automatic-Image-Labelling.git (accessed on 1 September 2021).

Acknowledgments: Thanks to NVidia GPU grant program for its support in providing GPU for free.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *39*, 640–651.
2. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2961–2969. [CrossRef]
3. Abousaleh, F.S.; Lim, T.; Cheng, W.-H.; Yu, N.-H.; Hossain, M.A.; Alhamid, M.F. A novel comparative deep learning framework for facial age estimation. *EURASIP J. Image Video Process.* **2016**, *2016*, 47. [CrossRef]
4. Ma, X.; Geng, J.; Wang, H. Hyperspectral image classification via contextual deep learning. *EURASIP J. Image Video Process.* **2015**, *2015*, 20. [CrossRef]
5. Li, X.; Jiang, Y.; Chen, M.; Li, F. Research on iris image encryption based on deep learning. *EURASIP J. Image Video Process.* **2018**, *2018*, 126. [CrossRef]
6. Xin, M.; Wang, Y. Research on image classification model based on deep convolution neural network. *EURASIP J. Image Video Process.* **2019**, *2019*, 40. [CrossRef]
7. Shi, W.; Liu, S.; Jiang, F.; Zhao, D.; Tian, Z. Anchored neighborhood deep network for single-image super-resolution. *EURASIP J. Image Video Process.* **2018**, *2018*, 34. [CrossRef]
8. Yang, W. Analysis of sports image detection technology based on machine learning. *EURASIP J. Image Video Process.* **2019**, *2019*, 17. [CrossRef]
9. Russa kovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
10. Qin, X.; He, S.; Zhang, Z.; Dehghan, M.; Jagersand, M. ByLabel: A Boundary Based Semi-Automatic Image Annotation Tool. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1804–1813. [CrossRef]
11. Iakovidis, D.K.; Goudas, T.; Smailis, C.; Maglogiannis, I. Ratsnake: A Versatile Image Annotation Tool with Application to Computer-Aided Diagnosis. *Sci. World J.* **2014**, *2014*, 286856. [CrossRef] [PubMed]
12. Chaudhary, A.; Raheja, J.L. Light invariant real-time robust hand gesture recognition. *Optik* **2018**, *159*, 283–294. [CrossRef]
13. McConnell, R.K. Method of and Apparatus for Pattern Recognition. U.S. Patent 4,567,610, 28 January 1986.
14. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 886–893. [CrossRef]
15. Sinha, S.N.; Frahm, J.M.; Pollefeys, M.; Genc, Y. GPU-based video feature tracking and matching. In *EDGE, Workshop on Edge Computing Using New Commodity Architectures*; Department of Computer Science: Chapel Hill, NC, USA, 2006.
16. Dutta, A.; Gupta, A.; Zisserman, A. Vgg Image Annotator Via. 2016. Available online: https://www.robots.ox.ac.uk/~{}vgg/software/via/ (accessed on 20 January 2022).
17. Dutta, A.; Zisserman, A. The VIA annotation software for images, audio and video. *arXiv* **2019**, arXiv:1904.10699.
18. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
19. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. In Proceedings of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981.
20. Tomasi, C.; Kanade, T. Detection and Tracking of Point Features. *Int. J. Comput. Vis.* **1991**, *9*, 137–154. [CrossRef]
21. Shi, J. Good features to track. In Proceedings of the 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994.
22. Breheret, A. Pixel Annotation Tool. 2017. Available online: https://github.com/abreheret/PixelAnnotationTool (accessed on 20 January 2022).

23. Zhang, C.; Loken, K.; Chen, Z.; Xiao, Z.; Kunkel, G. Mask editor: An image annotation tool for image segmentation tasks. *arXiv* **2018**, arXiv:1809.06461.
24. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic data for text localisation in natural images. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2315–2324.
25. Sun, B.; Saenko, K. From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. *BMVC* **2014**, *1*, 3. [CrossRef]
26. Su, H.; Qi, C.R.; Li, Y.; Guibas, L.J. Render for CNN: Viewpoint estimation in images using CNNC trained with rendered 3d model views. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2686–2694.
27. Castro, E.; Ulloa, A.; Plis, S.M.; Turner, J.A.; Calhoun, V.D.; Eduardo, C. Generation of synthetic structural magnetic resonance images for deep learning pre-training. In Proceedings of the 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), New York, NY, USA, 16–19 April 2015; pp. 1057–1060. [CrossRef]
28. Segawa, Y.; Kawamoto, K.; Okamoto, K. First-person reading activity recognition by deep learning with synthetically generated images. *EURASIP J. Image Video Process.* **2018**, *2018*, 33. [CrossRef]
29. Parkhi, O.M.; Vedaldi, A.; Zisserman, A.; Jawahar, C.V. Cats and dogs. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3498–3505. [CrossRef]
30. GoogleResearch. Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: https://doi.org/10.1207/s15326985ep4001 (accessed on 1 December 2018).
31. GoogleResearch. Detection Model Zoo. 2017. Available online: https://github.com/tensorflow/models (accessed on 20 January 2022).
32. Erol, A.; Bebis, G.; Nicolescu, M.; Boyle, R.D.; Twombly, X. Vision-based hand pose estimation: A review. *Comput. Vis. Image Underst.* **2007**, *108*, 52–73. [CrossRef]
33. Abderrahmane, Z.; Ganesh, G.; Crosnier, A.; Cherubini, A. Haptic Zero-Shot Learning: Recognition of objects never touched before. *Robot. Auton. Syst.* **2018**, *105*, 11–25. [CrossRef]