A Method for Training Object Scale Estimation System using Feature Extraction Enhancement with Depth Estimation

Kyungryun Kim

Detroit Country Day Upper School

ABSTRACT

In recent years, machine learning-based object scale estimation has been growing in popularity, as the significance of the technology lies in its potential for use in many industry fields. Although several methods have been proposed, the possible applications of this technique are limited due to its insufficient accuracy. Hence, a human-level accurate system is needed for the technology to be applied in the real-world domain. This research paper proposes a novel object scale estimation system that incorporates the feature extractor, disentangled feature maps, depth estimator, object localizer, and ground truth depth map. The input of the proposed system is an image, which is inputted into the feature extractor to create disentangled feature maps. These feature maps are then extracted by the depth estimator to generate a depth map, and by the object localizer to create a predicted bounding box around the object. The trained feature extractor can extract disentangled size-related features from the inputted image by jointly training the depth estimator and object localizer. The use of disentangled features boosts the performance of the proposed system. In addition, we propose an actual scale converter module to calculate the actual size of the input object. Throughout the experiments, the proposed method has proven that it is superior compared to other state-of-the-art methods. The proposed method achieves an IoU (Intersection over Union) value of 0.8113 on the COCO dataset.

1. Introduction

1.1 Problem Definition

The object scale estimation system aims to measure the dimension of a certain object in each input image. In recent years, the use of object scale estimation systems has been rapidly expanding in many fields, including smart farms or augmented reality (Loresco, Pocolo James, et al. 2018). However, the proposed system still lacks precision and generates numerous errors. For this reason, a human-level estimation system is a necessity for the object scale estimation system to be further applied in the real-world domain.

1.2 Naïve Approach

In early research, the majority of object scale estimation studies developed a way to directly estimate object dimensions. These methods demand a large-scale dataset in order to achieve adequate accuracy. However, collecting large-scale datasets is impractical because labeling the dimensions of each object is costly and time-consuming.

1.3 Relation-based Method

To solve the aforementioned problem, a two-stage approach has been proposed. The approach divides the object scale estimation process into object localization and actual dimension-converting processes. Object localization includes the production of a predicted bounding box on an object. The actual dimension-converting process is then applied by using a proportional relationship between a reference object and the actual object that the proposed system aims to convert. However, the use of the novel two-stage approach still has problems. The precision of the system still lacks compared to human-level precision. In addition, the error in generating the bounding box can propagate to the actual dimension-converting process which degrades the accuracy of the result.

1.4 The Proposed Method

In this paper, I propose a novel strategy to train the scale estimation system. The proposed system is composed of an encoder, a decoder, and a scale regression MLP (Multilayer Perceptron). One of the components, the encoder, takes the image as input and outputs feature maps that contain important image features. The decoder then takes in the feature maps that are more important for making an accurate depth map and an accurate scale estimator. Then, the scale regression MLP finally produces the scale of the object. I also demonstrate the proposed system on the application to edge devices such as Raspberry Pi. (Upton, et al. 2014)

2. Related Work

2.1 Representation Learning

Representation learning aims to automatically learn to find the representations for specific features that are contained in the dataset samples. In the deep learning approach, representation learning is often conducted via an autoencoder which is composed of an encoder and a decoder. The encoder takes input images and outputs features or latent code. These features contain characteristics of input images such as background, brightness, or image shape. The decoder takes these features as input and reconstructs the original input image. By jointly training these two networks, the encoder is enforced to extract important features in order to make the decoder accurately reconstruct the image. Figure 1 shows an autoencoder architecture and its representation learning pipeline.

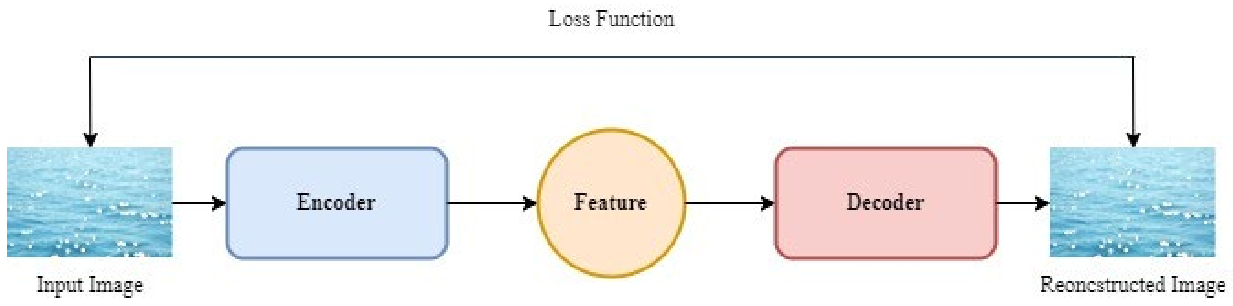

Figure 1. Example of autoencoder-based representation learning

In this paper, I exploit the representation learning approach to find better representation for accurate object scale estimation networks. I set figure 1. as a baseline and propose a novel representation training method using a depth estimator for accurate object scale estimation networks. A detailed explanation is in a future study in chapter 4.

2.2 Depth Estimation

Given input RGB color image, depth estimation aims to predict a depth map or a 3D reconstruction of the input image to find their distances from the camera in pixels. Traditional approaches are developed based on multi-view cameras to find the physical relationship between the multi-view images. Recent methods can directly estimate depth maps from an image captured by a monocular camera. This approach is often divided into two CNN (Convolutional Neural Network) modules which are an encoder and a decoder, and the depth map estimator.

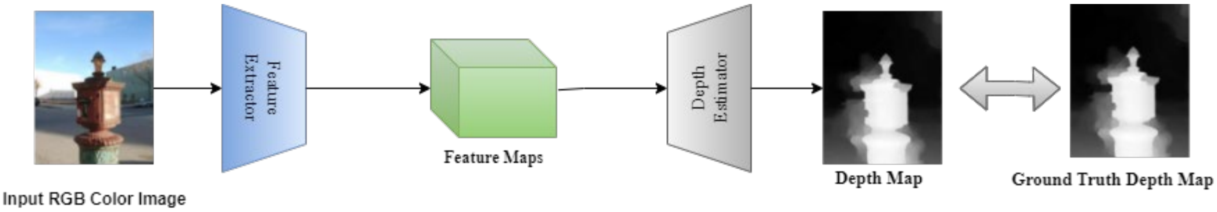

Figure 2. Example of depth estimation

Figure 2 represents the fundamental approach of the recent depth estimation methods. As the decoder is trained to estimate an accurate depth map, the encoder is forced to extract the depth-related features, or object shape-related features among entangled image features such as color histogram, illumination conditions, background, noise, or shape of the image. The proposed method extensively exploits these characteristics to capture the object shape-related features in order to accurately predict the bounding box of the objects. The underlying hypothesis here is that if the encoder can be trained to extract and disentangle the object shape-related features, it will positively affect the final accuracy of the trained method. In the next chapter, the detailed layout of this approach is explained in detail.

3. Method

3.1 Overview

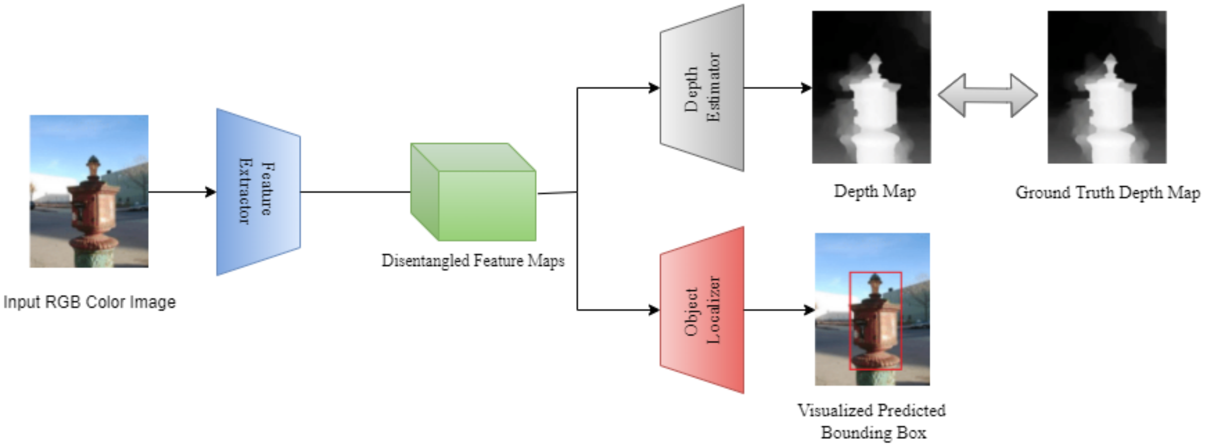

Figure 3. The overall architecture of the proposed object scale estimation system.

Figure 3 shows the overall architecture of the proposed scale estimation system. The proposed system consists of a feature extractor, depth estimator, object localizer, and scale convert module. Firstly, the feature extractor takes the object image I, and then generates feature maps F. These feature maps are then fed to the depth estimator and the object localizer. Using these feature maps, the depth estimator estimates depth maps $\widehat{D}$ which is compared with its corresponding ground truth D_{gt} later. The feature maps are also fed to the object localizer to produce the bounding box of the object, $\hat{B}$, as shown in Figure 1. Finally, the proposed scale convert module converts the pixel-level scale of an object calculated from the bounding box into the actual scale's unit such as an inch or centimeter.

Table 1. Notations used in this paper

Original	**Abbreviated**
Color input image	I
Feature maps	F
Estimated depth map	D
Ground truth depth map	D_{gt}
Predicted bounding box	B
Ground truth bounding box	B_{gt}

3.2 Scale Convert Module

The predicted bounding box implies the pixel-level scale of the object. The proposed scale convert module can easily convert the pixel-level scale into the actual scale unit. The actual scale of the object is calculated via proportional relationships between the pixel-level scale and the reference.

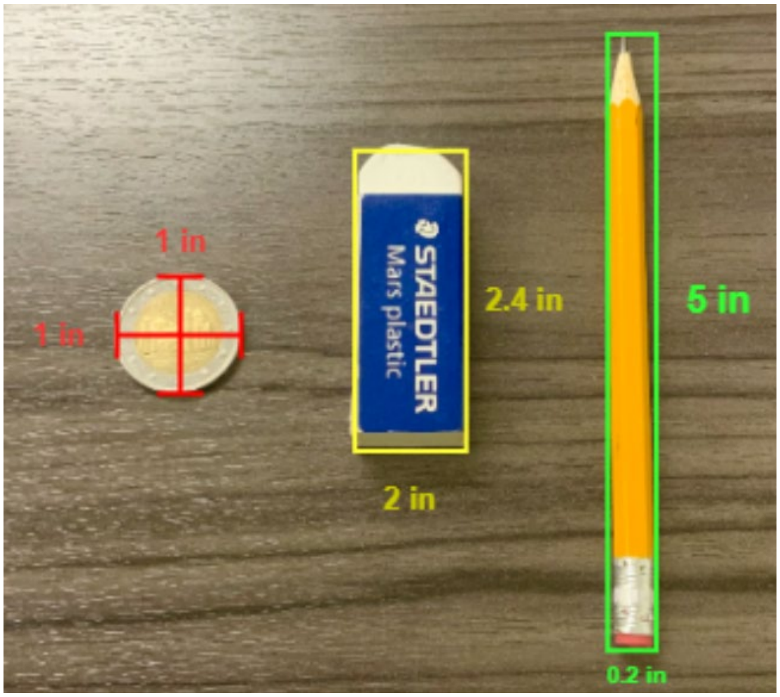

Figure 4. Example of the reference object and actual size converting process.

Figure 4 represents an example of an actual size-converting process with the reference and target objects. In this paper, the quarter is used as a reference object due to its common use, and symmetry, and is easy to calculate since its diameter is 1 inch. For example, to find the actual length of the eraser which is the target object (yellow bounding box), the pixel length is needed, which is calculated using the predicted bounding box from the aforementioned object localizer. The converting process is done by Equation (1) below.

Equation 1: Actual size converting process

$$S_{target} = \frac{S_{ref} * P_{target}}{P_{ref}}$$

Where P_{ref} and S_{ref} are the pixel size and actual size of the reference object (a quarter coin in figure4). P_{target} and S_{target} denote the pixel size and actual size of the target object (yellow and green box object in figure 4). As the object localization network outputs, the bounding box of each object the P_{ref} and P_{target} can be easily obtained. The variable S_{target} can be derived from equation 1 since S_{ref} is provided as prior knowledge (the diameter of a quarter coin is always 1 inch).

3.3 Loss Function

In this paper, I use two kinds of loss functions to train the proposed object scale estimation system. To train the feature extractor and the depth estimator, I use MSE (Mean Squared Error) loss function which is commonly used for depth estimation research (Khan, et al. 2020), (Zhao, Chaoqiang, et al. 2020). More importantly, it is used to calculate the average pixel-wise difference in feature maps by using summations. The MSE loss function is defined as Equation (2).

Equation 2: Mean squared error

$$L_{mse} = \frac{1}{W*H}\sum_{i=1}^{W}\sum_{j=1}^{H}(D(xi-yj),D_{gt}(xi-yj))^2$$

Where, *W* and *H* denote the width and height of the image, which is used to calculate the average pixel-wise difference in the predictions and its corresponding ground truths. The pixel-wise differences are then squared and summated and averaged.

3.4 Architecture

To develop the proposed feature extractor, I exploit imagenet pretrained (Krizhevsky, et al. 2017) resnet18 (He, Kaiming, et al. 2016) architecture without final linear layers. Through extensive experiments, I have found that resnet18 shows the best performance and accuracy among the existing off-the-shelf convolutional neural networks such as VGG (Simonyan, et al. 2014), DenseNet (Huang, Gao, et al. 2017), and HRNet (Wang, Jingdong, et al. 2020). The architecture of the proposed depth estimator is also heavily based on resnet. It has upsampling layers between each ResBlock in the vanilla resnet to reconstruct the original resolution of the input image. I simply implement two linear layers to build the proposed object localizer. Heuristically, the non-linearity power of the object localizer is enough to produce accurate results.

3.5 Implementation Details

I used Adam with an initial learning rate of 0.0001 and a decay rate of 0.1. The learning rate will decay first at 80 epochs, and then at 160. I also used 16 batches and 200 epochs for training the model. The model is also trained using data augmentation. Specifically, the two techniques in data augmentation are horizontal flip and color jitter (the change of colors in the image).

4 Experimental Results

In this chapter, I explain how the experiment is conducted to prove the superiority of the proposed method.
The detailed evaluation protocol will be explained including the datasets, evaluation metrics, and comparison methods.

4.1 Dataset

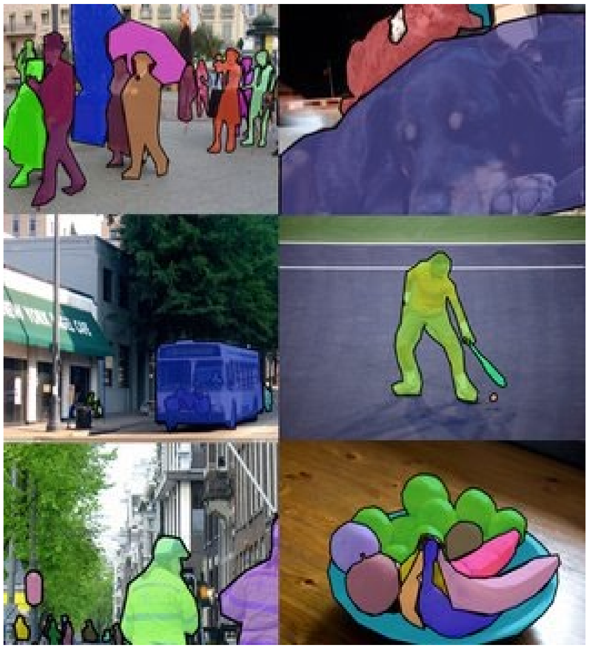

Figure 5. Example samples of COCO dataset (Liu, Tsung-Yi, et al. 2014)

For the dataset, I used COCO (Liu, Tsung-Yi, et al. 2014) object detection data that is frequently used to train object detection or location networks. The dataset contains 200,000 image samples of test sets, validation, and training. In addition, there are 80 object categories in the dataset, and it consists of high-quality images which allow better results for my proposed system. Figure 5 shows a snippet of the dataset which contains various types of object categories.

4.2 Evaluation metric

To evaluate the effectiveness of the proposed model, I used the IoU (Intersection over Union) metric which is widely known for its ability to measure the reliability or the effectiveness of object detection or object localization networks. The IoU metric output values from 0 to 1 and is calculated by dividing the area of intersection and the area of union.

Figure 6. Shows the process of calculating the IoU value, which shows the difference between the predicted and the actual bounding box. The numerator is composed of the area of the intersection (shaded part) and the denominator is the area of union, which is the area of the shaded part in the numerator.

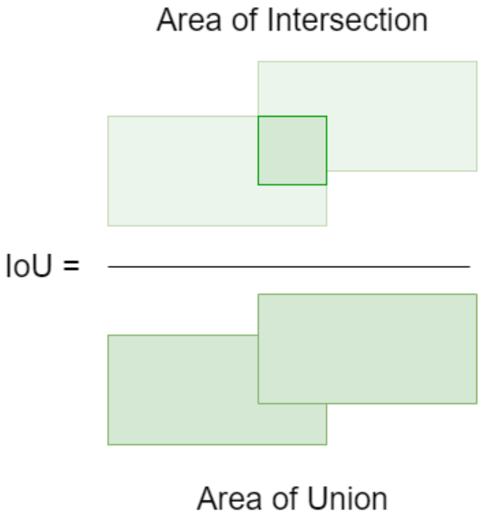

Figure 6. Example of IoU calculation

4.3 Comparison with the state-of-the-art methods

Table 2. IoU comparison to the previous State-of-the-art methods

	IoU
SSD (Liu, Wei, et al. 2016)	0.7296
Faster R-CNN (Ren, Shaoqing et al. 2015)	0.7784
Ours	0.8113

For the comparison methods, I choose SSD (Liu, Wei, et al. 2016) and Faster R-CNN (Ren, Shaoqing et al. 2015) which show comparable performance in object localization tasks. As shown in Table 2, the IoU value for the state-of-the-art methods 1 and 2 both respectively achieved 0.7296 for method 1 and 0.7784 for method 2. In comparison, the proposed method achieved an IoU value of 0.8113. The proposed method, compared to the state-of-the-art method 1 has an increase of 0.0817 in IoU value. The proposed method also outperforms state-of-the-art method 2 by 0.0329 IoU value.

The proposed method in comparison to the state-of-the-art techniques was able to achieve higher IoU values because of the joint training of both the depth estimator and the object localizer. Since the networks are jointly trained, the networks try to reduce the loss values of the depth estimator and the object localizer. This unique training strategy allows the trained model to produce more accurate localization results since the trained depth-aware encoder extracts richer features. The effectiveness of the proposed training strategy is proved in chapter 4.4.

4.4 Ablation study

Table 3. Ablation study results

Method	IoU
Ablation model	0.7804
Full model	0.8113

In this experiment, I evaluate the effectiveness of the proposed joint training strategy. First, I train the model using the proposed training strategy referred to as the full model. For the comparison, I also trained an ablation model without the depth estimator. Note that the ablation model only includes the object localizer to generate the bounding box. The full model as compared to the ablation model has an increase of 0.0309 IoU value. By analyzing the experimental results as shown in Table 3, it can be concluded that the proposed method helps with generating a more accurate output.

4.5 Additional experiments on common objects

In this chapter, I conduct an additional experiment to prove how the object localizer's increase in performance can affect the precision of the size-converting process. To measure the size of the object, I use 50 common objects such as laptops, books, pencil pouches, phones, binders, keyboards, mice, bags, toothbrushes, shirts, helmets, wallets, blankets, paper, tables, shoes, rulers, drawers, doors, light switches. The collected image samples are then fed to the proposed object localizer to produce bounding boxes for each object. The following process is the same as in the aforementioned operation explained in chapter 4.2. In table 4, I calculate the error in width and height of previous state-of-the-art methods and compare it to the proposed system.

Table 4. Comparison results on common objects

	Width error (inch)	Height error (inch)
SSD (Liu, Wei, et al. 2016)	2.63	3.18
Faster R-CNN (Ren, Shaoqing et al. 2015)	3.52	3.09
Ours	2.33	2.87

Since our proposed object localization system is superior to the existing state-of-the-art methods, the error propagation must also be lower than the methods used in previous research. Table 3 and 4 clearly show that the proposed method outperforms not only in the object localization but also in actual size estimation.

4.6 Application

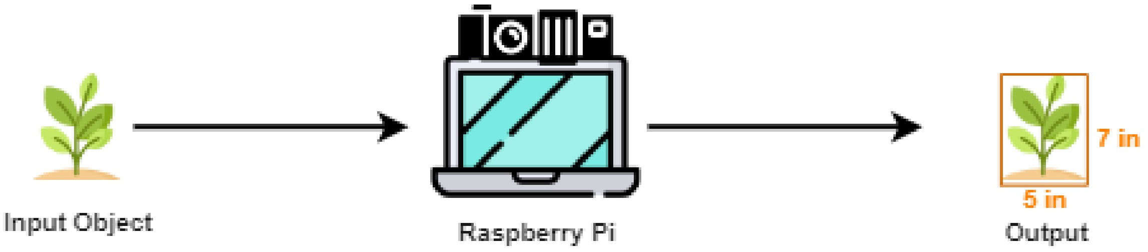

Figure 7. Real World Application

Finally, I will explain how I expanded the proposed method to the real world. Figure 7 displays a diagram of the application that I implemented on edge devices such as Raspberry Pi (Upton, et al. 2014). As shown in the figure above, the webcam captures a picture of the plant object, and then the Raspberry Pi activates the scale estimator to output the actual size of the input plant object. This application can be applied to the smart farm industry and allows the automation of the farming process. As the system can measure the size of the plants, it can adjust the essential factors for the plants such as the water pump or lightning strength with the estimated plants' life cycle. The software I used to implement the system is PyTorch (Paszke et al. 2017).

5 Conclusion

In this research project, I proposed a novel training approach for an object scale estimation system. The proposed approach comprises an encoder, depth estimators, and object localizers. The encoder takes in the input images and creates feature maps. These feature maps are inputted into the depth estimator and the object localizer. The depth estimator generates a depth map, and the object localizer predicts a bounding box. I also proposed the actual size converting process to accurately convert the bounding box size into inches which is a common metric unit for measuring size. To evaluate the effectiveness of the proposed method, I conducted comparison experiments with previous state-of-the-art methods. The proposed method had an increased IoU value of 0.0817 compared to the state-of-the-art method 1 and 0.0329 for method 2. It is proven that the proposed idea affected the results via an ablation study. I also conducted additional experiments on common objects to verify the proposed method outperforms the previous methods in the size-converting process as well. The extensive experiment results clearly show that the proposed method outperforms not only object localization but also actual size estimation. In the future, I plan to apply the proposed method to real-world problems such as the smart farm industry. I expect that the proposed method can allow the fully-automation of the farming process.

References

Eigen, David, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network." *Advances in neural information processing systems* 27 (2014).

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Huang, Gao, et al. "Densely connected convolutional networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Joglekar, Apoorva, et al. "Depth estimation using monocular camera." *International journal of computer science and information technologies* 2.4 (2011): 1758-1763.

Khan, Faisal, Saqib Salahuddin, and Hossein Javidnia. "Deep learning-based monocular depth estimation methods—A state-of-the-art review." *Sensors* 20.8 (2020): 2272.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.

Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *European conference on computer vision*. Springer, Cham, 2014.

Liu, Wei, et al. "Ssd: Single shot multibox detector." *European conference on computer vision*. Springer, Cham, 2016.

Loresco, Pocholo James, et al. "Computer vision performance metrics evaluation of object detection based on Haar-like, HOG and LBP features for scale-invariant lettuce leaf area calculation." *Int. J. Eng. Technol* 7.4 (2018): 4866-4872.

Paszke, Adam, et al. "Automatic differentiation in pytorch." (2017).

Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems* 28 (2015).

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014)

Upton, Eben, and Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

Wang, Jingdong, et al. "Deep high-resolution representation learning for visual recognition." *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020): 3349-3364.

Xu, Dan, et al. "Structured attention guided convolutional neural fields for monocular depth estimation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Zhao, Chaoqiang, et al. "Monocular depth estimation based on deep learning: An overview." Science China Technological Sciences 63.9 (2020): 1612-1627.