



Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# Depth estimation from monocular images

Master's Thesis
Master in Advanced Telecommunication Technologies

**Author:** Jordi Morera i Trujillo
**Advisors:** Ramon Morros Rubió, Javier Ruiz Hidalgo

**Universitat Politècnica de Catalunya (UPC)**
**2019 - 2020**

# Abstract

During this project, state-of-the-art deep learning models have been used to estimate depth maps from a monocular RGB image applying a teacher-student learning approach.

This paradigm has been used in order to distillate the knowledge of high capacity deep neural networks into shallower ones to make inference faster for real-time applications.

Some successful applications of this technique can be found both at natural language and computer vision applications.

# Acknowledgements

I would like to express my special thanks of gratitude to both supervisors for the dedication and the amount of ideas proposed for this project.

Secondly I would also like to thank Fran Roldán for his intuition at finding problems and proposing methods to correct them.

Finally I want to give my thanks of gratitude to all members of the Image Processing Group, specially Albert Gil Moreno and Josep Pujal Suria for their incredible work at managing the Calcula computing platform.

# Revision history and approval record

| Revision | Date | Purpose |
|---|---|---|
| 0 | 27/12/2019 | Document creation |
| 1 | 22/01/2020 | Document revision |
| 2 | 25/01/2020 | Document approbation |

DOCUMENT DISTRIBUTION LIST

| Name | e-mail |
|---|---|
| Jordi Morera | jordi.morera.trujillo@alu-etsetb.upc.edu |
| Ramon Morros Rubio | ramon.morros@upc.edu |
| Javier Ruiz Hidalgo | j.ruiz@upc.edu |

| Written by: | | Reviewed and approved by: | | Reviewed and approved by: | |
|---|---|---|---|---|---|
| **Date** | 24/01/2018 | **Date** | 25/01/2020 | **Date** | 25/01/2018 |
| **Name** | Jordi Morera | **Name** | Ramon Morros | **Name** | Javier Ruiz |
| **Position** | Project Author | **Position** | Project Supervisor | **Position** | Project Supervisor |

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

When capturing an image, we are projecting the 3-dimensional scene seen by the objective of the camera into a 2D plane as seen on Figure 1.1. This projection loses the information of how far was each point from the camera. Those distances can be represented in single-channel images called depth maps like the one at Figure 1.2.

Depth maps can be made acquired using specialized hardware such as Microsoft's Kinect or estimated using using one or more RGB views. Predicting depth is an essential component in understanding the 3D geometry of a scene. Making such estimation from stereo images using local correspondences between both views is a well-defined problem but when having one single view of the scene the problem becomes less straightforward and more ambiguous.

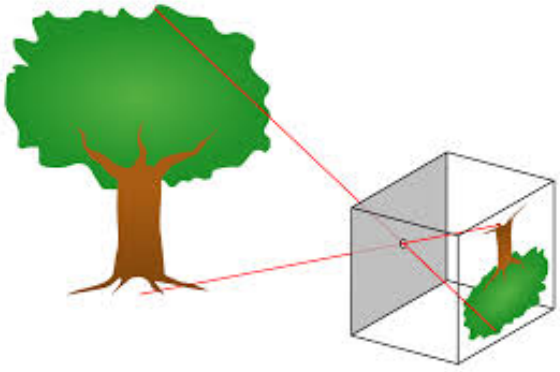


Figure 1.1: Pinhole camera model

Depth maps are mostly used in 3D applications such as modelling 3D shapes, rendering of 3D scenes more efficiently or shadow mapping. Furthermore, the use of depth information can also improve the performance of other computer vision tasks such as semantic segmentation[25]

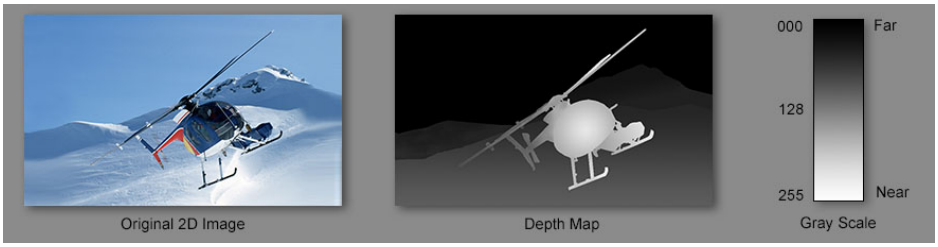



Figure 1.2: RGB Image and 8-bit depth map

## 1.1 Motivation

The motivation of this project is to implement a depth map estimation system applying different machine learning paradigms and compare the obtained results with the state-of-the-art

papers on this active research topic.

## 1.2 Requirements and specifications

The requirements of this project are the following:

- Implementing and training deep learning architectures to estimate depth maps from monocular images.
- Exploring different learning approaches such as student-teacher learning.
- Producing reproducible results by following machine learning good practices.
- Applying design patterns to produce low-coupled reusable software

As this is a research project there is no strict specification but obtaining results as good as possible.

## 1.3 Methods and procedures

This work is a continuation of the Introduction to Research project made by the author of this thesis in which a Pyramid Scene Parsing Network[26] was used for depth prediction on the ScanNet Dataset[8].

Both qualitative and quantitative results were good but we were not able to compare them with other researchers because most of the publications use other datasets to benchmark its performance.

## 1.4 Work Plan

This project has been developed by GPI research group at Universitat Politècnica de Catalunya, having a regular weekly meeting between supervisors and author to discuss decisions to be made. The work plan is described in the following work packages and Gantt diagram, as well as the modifications introduced since the first version.

### 1.4.1 Work Packages

- WP 0: State-of-the-art review.
- WP 1: Data acquisition and management.
- WP 2: Implementation of the network and training code.
- WP 3: Experimentation

- WP 4: Comparison of results with the literature.
- WP 5: Documentation

### 1.4.2 GANTT Diagram

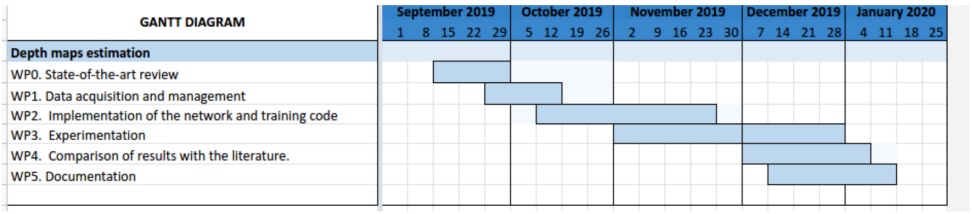



Figure 1.3: GANTT diagram followed during the project

## 1.5 Incidents and Modifications

Data acquisition work package took longer than expected due to the need for temporal synchronization and spatial alignment between the depth sensor and the camera data.

Another problem faced during the training of the networks is numerical instability. It could appear in any phase of the training making the gradients explode or not even appear at all while running the same code.

# Chapter 2

# State of the art

Depth maps estimation from monocular images state of the art methods is also used for other similar computer vision tasks such as semantic segmentation because of the similarity of both problems and their inner difficulties.

In this kind of computer vision tasks CNN (Convolutional Neural Networks) excel at extracting useful contour and texture features and making use of them for classification, segmentation, contour detection and other tasks.

The main problem those networks face in dense prediction tasks is the coarse outputs, it consists on a noticeable lack of detail usually around the contours of the predicted image as seen on Figure 2.1.

Both supervised and unsupervised machine learning[27] have been applied to the topic. As we will be approaching this problem from a supervised learning perspective I will focus on those methods.

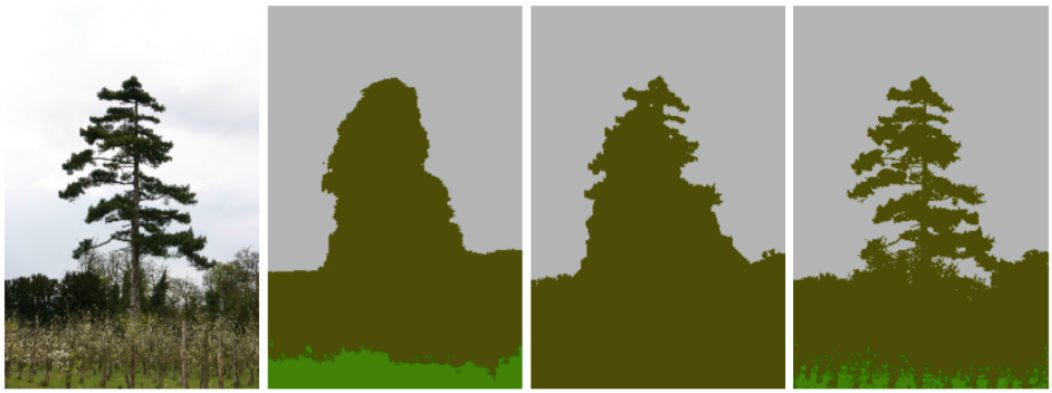


Figure 2.1: Coarse to fine semantic segmentation

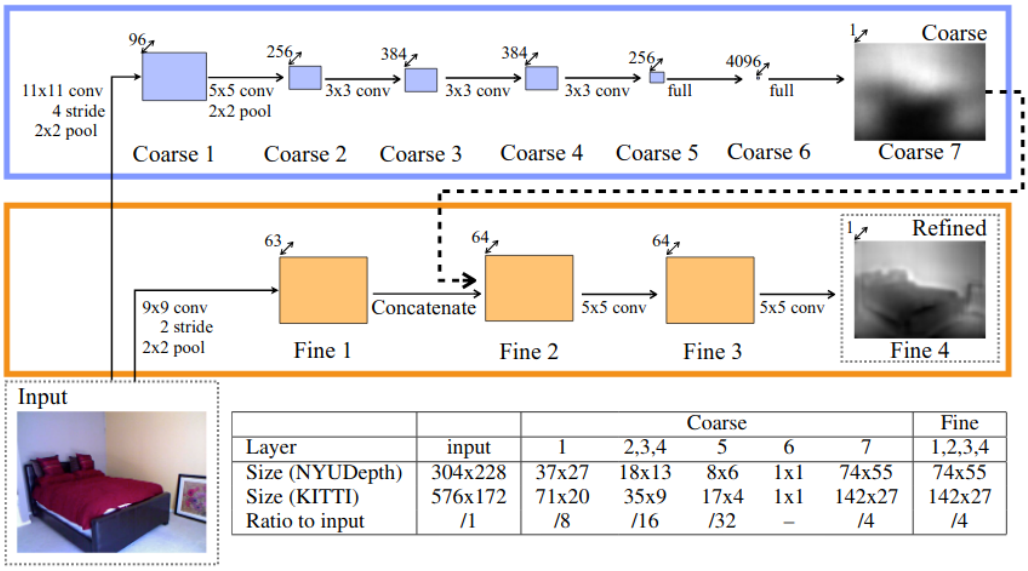
## 2.1 Supervised depth estimation from monocular images

### 2.1.1 Multi-scale methods

An intuitive manner of minimizing the coarse outputs problem is using the image at different resolutions. This way both the global context of the image and its details can be better used for the estimation. Furthermore, a more efficient way to obtain similar results with a much lower amount of operations and parameters is using feature maps at different resolutions.

The approach presented at Depth Map Prediction from a Single Image using a Multi-Scale Deep Network [9] uses one network to make an initial coarse estimation, this estimation is used as

a feature map in the second network which refines the initial estimation with higher-level details. As seen in Figure 2.2 last layers of the Coarse estimation block are fully-connected, this decision was made to make a better use of the global context of the image but increases a lot the amount of parameters of the network making it not suitable for soft real time applications.




| | | Coarse | | | | | Fine |
|---|---|---|---|---|---|---|---|
| Layer | input | 1 | 2,3,4 | 5 | 6 | 7 | 1,2,3,4 |
| Size (NYUDepth) | 304x228 | 37x27 | 18x13 | 8x6 | 1x1 | 74x55 | 74x55 |
| Size (KITTI) | 576x172 | 71x20 | 35x9 | 17x4 | 1x1 | 142x27 | 142x27 |
| Ratio to input | /1 | /8 | /16 | /32 | – | /4 | /4 |

Figure 2.2: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network architecture

The U-Net architecture presented at Convolutional Networks for Biomedical Image Segmentation [21] is fully-convolutional and contains no fully connected layer.The main architectural change in the U-Net is the use of skip connections between mirrored layers of an encoder-decoder scheme as seen in Figure 2.3 This characteristic let the decoder see feature maps of different resolutions while up-sampling the lower-dimensional representations making it easier to reconstruct fine details of the image. This architecture has also been used in and adversarial manner at Image-to-Image Translation with Conditional Adversarial Networks[14].

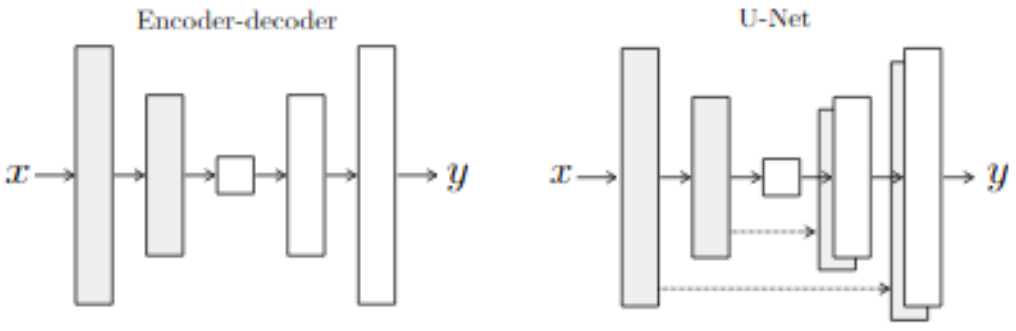



Figure 2.3: Comparison between encoder-decoder architecture and U-Net

### 2.1.2 Pyramid Pooling

Pyramid Scene Parsing Network[26] got the champion of ImageNet Scene Parsing Challenge 2016 with an innovative approach for exploiting the global context of an image called Pyramid Pooling.

Pyramid Pooling modules objective is making efficient use of the global context of the image for pixel-level prediction tasks. It is achieved by applying different-region-based context aggregation to each feature map, up-sampling them and combining them into the prior representation. This prior representation is then concatenated with the original feature maps and forwarded throw the decoder as seen in Figure 2.4.

An example of the use of pyramid pooling in depth map prediction can be found at Structure-Aware Residual Pyramid Network for Monocular Depth Estimation[6]. This architecture achieves state-of-the-art performance in both qualitative and quantitative evaluation.

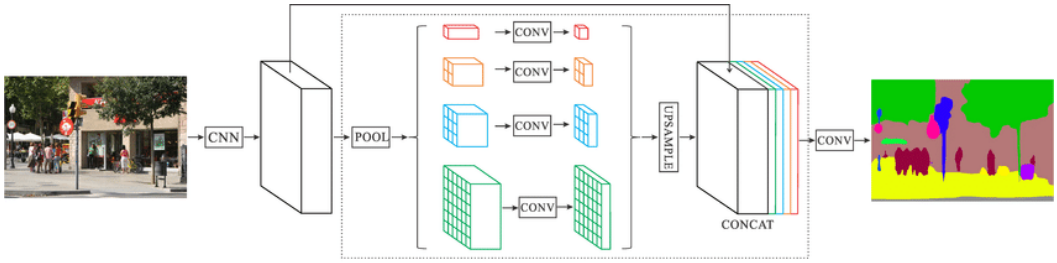



Figure 2.4: Pyramid Scene Parsing Network architecture.

### 2.1.3 Attention based methods

Attention mechanisms were proposed at Attention is all you need[23] for neural machine translation to properly handle large sequences by letting the network focus more on some words than others while decoding input sentences.

The same concept has also been applied in depth map estimation at Attention-based Context Aggregation Network for Monocular Depth Estimation[7]. The proposed approach deals with image context information with a pooling module while capturing pixel-level information with a self-attention module following the Figure 2.5 schema.

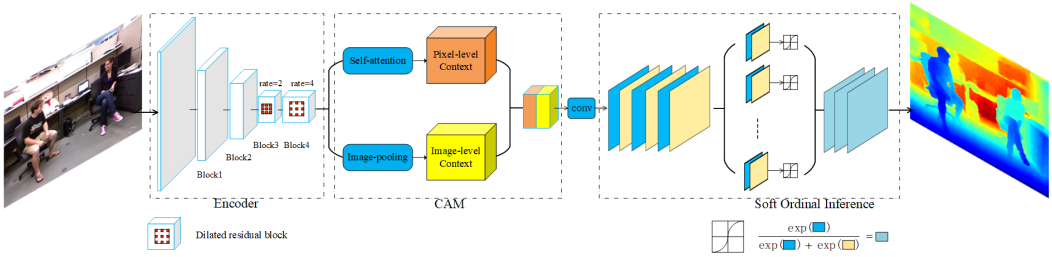



Figure 2.5: Attention-based Context Aggregation Network architecture.

# Chapter 3

# Methodology

This section will explain the methodologies followed to train a teacher model making use of privileged information and transfer this knowledge to a student network under the framework proposed at A Generalized Meta-loss function for regression and classification using privileged information[1].

This approach consists of making use of privileged information only available at training to develop a first model (teacher). This first model is then used giving hints to the learning of a student model which no longer uses privileged information but has the same behaviour.

The main actors of the framework applied for depth map estimation are:

1. Privileged information: A low dimensional and learnable representation of the depth maps will be used together with RGB image to improve the reconstruction of the depth maps.
2. Teacher network: Extracts features from both input RGB images and the target depth map and fuses them across a decoder to make the estimation. It's only goal is making depth reconstructions as good as possible.
3. Student network: Extracts features from monocular RGB images that mimic teacher's feature space and estimates depth with such features.

In order to measure the impact of this paradigm our student will be compared with a baseline model with the same architecture proposed at Figure 3.2 trained directly to map RGB images to depth maps following Section 3.2.1 methodology.

## 3.1 Model architecture

Networks trained in this project are based on U-Net architecture, this decision was made because of the wide adoption of that network for dense prediction tasks like image segmentation or image super-resolution[24].

### 3.1.1 Student network

A custom implementation of the U-Net architecture which replaces original convolutional blocks for residual ones has been used as student network. This way we explicitly reformulate the layers of the network to learn residual functions with reference to the layer inputs instead of learning unreferenced functions as seen in Figure 3.1 Comprehensive empirical evidence showing that these deep residual networks are easier to optimize than other architectures is provided at Deep Residual Learning for Image Recognition[12].

One more factor that influenced the previous modification is that it allows us to initialize the encoder of the model with Imagenet ResNet pre-trained weights instead of random initialization making training converge faster.

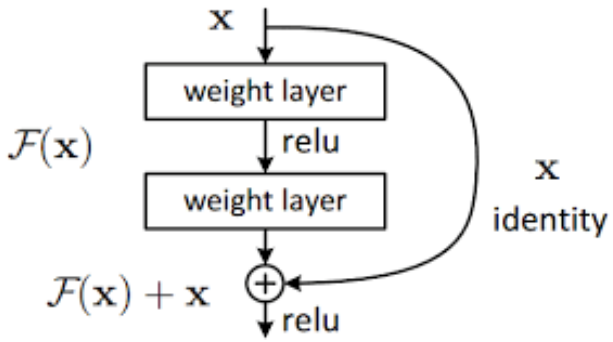



Figure 3.1: Building block for residual learning

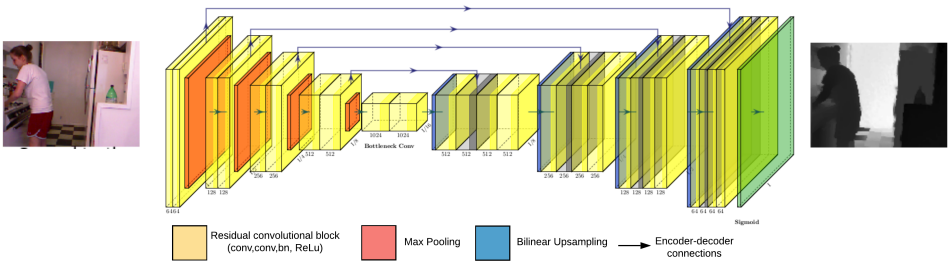



Figure 3.2: U-Net Architecture. Generated using PlotNeuralNet library and Latex code

### 3.1.2 U-Net teacher network: Learning with privileged information

The main challenge when designing our teacher is that it must work better than any other approach so that even if the student loses part of the performance it still behaves at least as good as our baseline model. In order to do so we had to give our teacher some advantage on learning this task and decided to let our decoder see part of the ground truth as a low dimensional representation of the depth map.

The proposed design of the teacher network consists of 2 residual encoders. The bottom one on Figure 3.3 extracts a low-dimensional representation of the depth map while the upper one extracts features from the RGB images at different resolutions. Finally, a decoder reconstructs the depth map from the depth manifold using also RGB features maps from the second encoder.

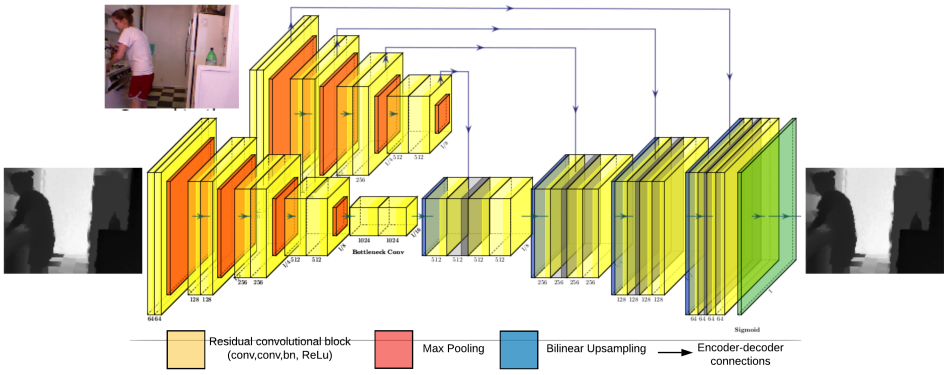



Figure 3.3: U-Net Teacher Architecture. Generated using PlotNeuralNet library and Latex code

## 3.2 Training methodology

Training a deep neural network involves lots of challenging tasks such as defining the loss function, choosing which gradient descent based optimization algorithm to use and prevent numeric instability problems such as vanishing and exploding gradients.

Optimization algorithms used:

- Adam[15]: Provides a fast convergence of the model weights as its best point but it is known to get stuck on local minima.
- SGD[20]: The model takes more iterations to converge than the previous algorithm but can find more optimal solutions if its hyper-parameters are suited for the geometry of the loss function.

To prevent over-fitting problems dropout will be applied at the decoder layers. Dropout[3] consists on randomly zeroing activations of a layer with a probability *p*. The theory behind dropout in neural networks is that removing neurons during training forces the network to learn redundant and more robust representations. However, Efficient Object Localization Using Convolutional Networks[22] states that this technique was not preventing over-fitting on fully-convolutional networks which has finally been our case.

### 3.2.1 Teacher training

Network is trained by minimizing the estimation errors with a loss function with 2 terms. The first term $L_{depth}$ consists on the root-mean-square error(RMSE) in log scale between the ground truth depth map *d* and the network estimation *p*, equation 3.1. Supervising in log scale makes the network focus more on closer objects.

The second term of the loss $L_{grad}$ is defined as the L1 norm between the gradient of the depth map and the gradient of the estimation, it penalizes errors round edges of the depth map minimizing the coarse outputs problem. Depth gradient-based losses have been widely used

for depth estimation as stated at Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries[13].

Gradients used at equation 3.2 have been computed using a 3x3 Sobel operator as $\nabla$.

$$L_{deph} = \frac{1}{n}\sum_{n=1}^{n}\sqrt{||log(d_i) - log(p_i)||^2} \tag{3.1}$$

$$L_{grad} = \frac{1}{n}\sum_{n=1}^{n}|\nabla d - \nabla p| \tag{3.2}$$

Finally, both terms are added with specific weights as seen in equation 3.3. The value of those weights was chosen so that both loss magnitudes were similar.

$$L_{total} = \alpha L_{Depth} + \beta L_{Grad} \tag{3.3}$$

### 3.2.2 Student training

Student network will benefit from the previously learned depth representations by mimicking the teacher. It is achieved by adding 2 new distillation terms to previous reconstruction loss.

Those terms, $L_{emb}$ and $L_{repr}$ at equations 3.4 and 3.5 are defined as the euclidean distance between teacher's and student's feature maps at different stages of the network. It can be seen from a Bayesian point of view as adding a prior about how student's feature maps should be from the teacher's knowledge. Feature maps are compared at the last layer of the encoder for $L_{emb}$ (after Bottleneck block in Figure 3.3) and before the last convolution of the decoder for $L_{repr}$ as seen in Figure 3.4

Blocks r1 and r2 are adaptation layers to better learn from intermediate teacher distributions improving hint learning as proposed at Learning Efficient Object Detection Models with Knowledge Distillation[4].

$$L_{emb} = \frac{1}{2}||r1(z_{e_rgb}) - z_{e_d+rgb}||^2 \tag{3.4}$$

$$L_{repr} = \frac{1}{2}||r2(rep_{e_rgb}) - rep_{e_d+rgb}||^2 \tag{3.5}$$

Previous losses are added together with the reconstruction ones following Equation 3.6

$$L_{Total} = \alpha L_{depth} + \beta L_{grad} + \gamma(L_{emb} + L_{repr}) \tag{3.6}$$

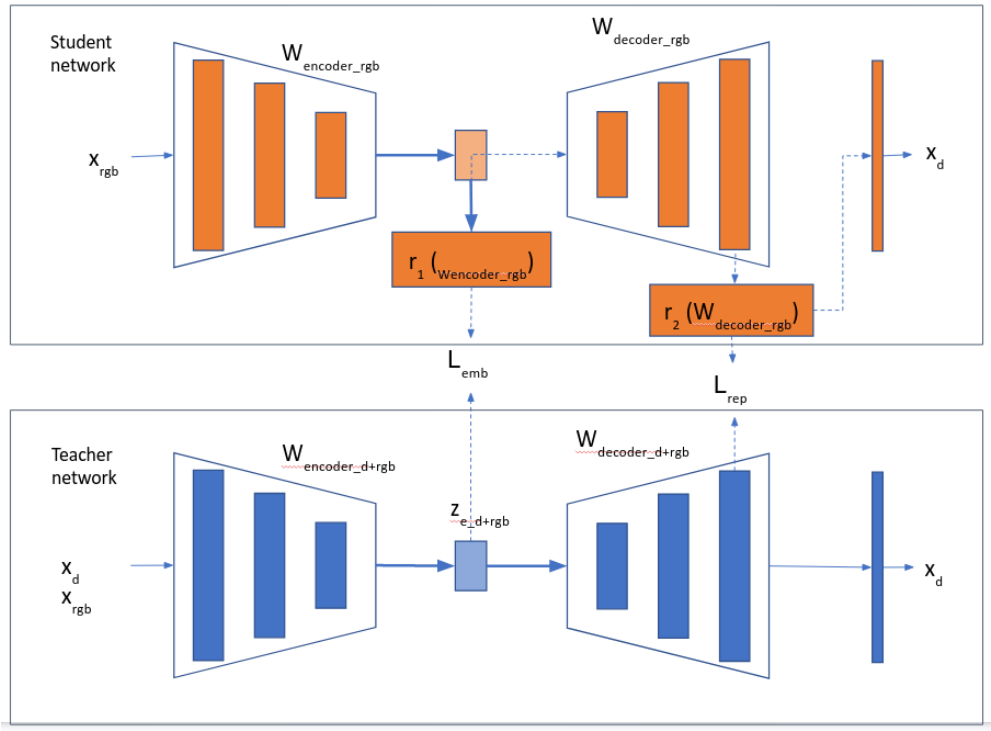



Figure 3.4: Distillation loss graphic example. The teacher architecture has been simplified and encoder-decoder connections are not represented in the image.

Steps followed to train student network:

1. Train teacher network following Section 3.2.1
2. Freeze teacher network
3. Train an student network minimizing both reconstruction and distillation losses
4. Use student network for inference

### 3.2.3 Using teacher without privileged information

An intuitive way of adapting our teacher architecture to infer depth maps without privileged information consists on freezing the RGB encoder and decoder of the teacher and replace the depth maps encoder with a new one that uses RGB information. This way we can keep most of the weights of our teacher unaltered while using domain adaptation to replace the privileged information for adapted RGB features.

This adaptation is implemented using a new RGB encoder that is trained to mimic teacher's depth encoder minimizing loss $L_{emb}$ seen at Section 3.2.2 . This approach can be summarized as adapting RGB images to fit the depth representation domain that our teacher already deals without modifying teacher decoder.

1. Train teacher network following Section 3.2.1

2. Freeze teacher network
3. Train a new encoder to adapt input images to the depth representations domain our decoder already deals with.
4. Infer without privileged information using the new encoder instead of the former one.

# Chapter 4

# Results

## 4.1 Computational requirements

Experiments have been run with the computational resources available at the Image Processing Group of the Universitat Politecnica de Catalunya.

GPI research group has a cluster of servers which is shared between all the research group and in which we ran our experiments. For each experiment the system reserves the amount of RAM, CPU cores and GPU requested by the user. If there are no available resources tasks are queued until resource is available.

## 4.2 Dataset: NYU Depth V2

NYU Depth V2 Dataset[19] consists of a set of interior scenes recorded by both an RGB camera and Kinect depth sensor.

A set of 1499 densely labelled pairs of aligned RGB and depth images is provided but in order to increase the size of the dataset, the original video sequences will be used. It implied the following data transformations:

- Temporal synchronization of both sensors because of different sampling frequencies.
- Projection of the image into the depth plane using the calibration parameters.
- In-painting missing pixels in the depth maps

The previous list of task was accomplished using NYU Toolkit and some Matlab programming.

The dataset obtained after reducing the temporal redundancy of the video by down-sampling a factor 25 (1 frame per second of video) consists on 590 different scenes containing a total of 42.320 aligned depth and RGB images which will be split into 3 partitions in order to cross-validate our solution and test it. To avoid data leaks between partitions we will split the scenes, this way our network will never be validated or tested with images from a scene it has already seen in the training phase.

| | Train | Validation | Test |
|---|---|---|---|
| Scenes | 354 | 157 | 79 |
| Images | 26.266 | 10.340 | 5.714 |

Table 4.1: Dataset partitions

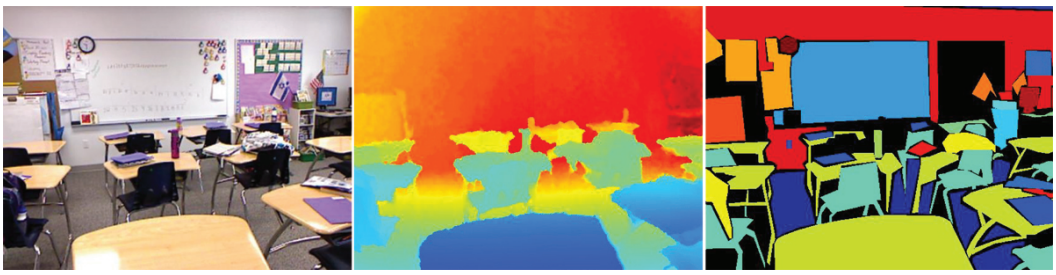


Figure 4.1: Output from the RGB camera (left), preprocessed depth (center) and a set of semantic labels (right) for the image.

## 4.3 Data pre-processing

When working with this amount of data and deep neural networks developing an efficient data processing pipeline become inevitable if the researcher wants to take profit of high-performance hardware such as GPUs.

The framework used to develop the projects (PyTorch) exposes some interfaces for implementing multi-threaded image data loaders which will be used to both read, normalize and augment the data.

- Data normalization: Neural networks are known to be quite sensitive to the scale and distribution of its inputs. The mean and the standard deviation of the training split of the dataset will be used to normalize our input data. Furthermore depth maps will be scaled between 0 and 1.
- Data augmentation: This technique consists on artificially expanding the size of a training dataset by creating modified versions of images. From a statistical point of view, this technique increases the variability of the data preventing overfitting problems. Figure 4.2 shows and example of data augmentation applied to both input image and segmentation masks.

Data augmentation pipeline:

- Crop: Randomly crop both image and depth map at the same coordinates to $\frac{3}{4}$ of its original shape.
- Flip: The RGB and the depth map are both horizontally flipped with 0.5 probability.
- Color Jitter: Randomly change the brightness, contrast and saturation of an image with 0.5 probability .


Figure 4.2: Image augmentation applied to both image and semantic masks

## 4.4 Software architecture

Code related to the development of this thesis has followed SOLID patterns so that is could be reused and easily modified/mantained. In order to do so we have used the Protected Variation pattern to make our system agnostic to the following change scenarios:

- Working with other datasets.
- Training different models.
- Learning other tasks such as classification or segmentation.

Figure 4.3 shows a UML diagram containing the most important classes of the project and how they are related.

- BaseDataset: Superclass which implements common methods for reading, processing and yelding data efficiently. Each new concrete dataset should inherit from BaseDataset and override specific methods according to the task/data needs.
- BaseModel: Provides useful methods such as extractFeatures, predict, updateWeights while being agnostic of the architecture of the model itself.
- Transformation: Interface to develop custom transformations compatible with the Albumentations Python library.

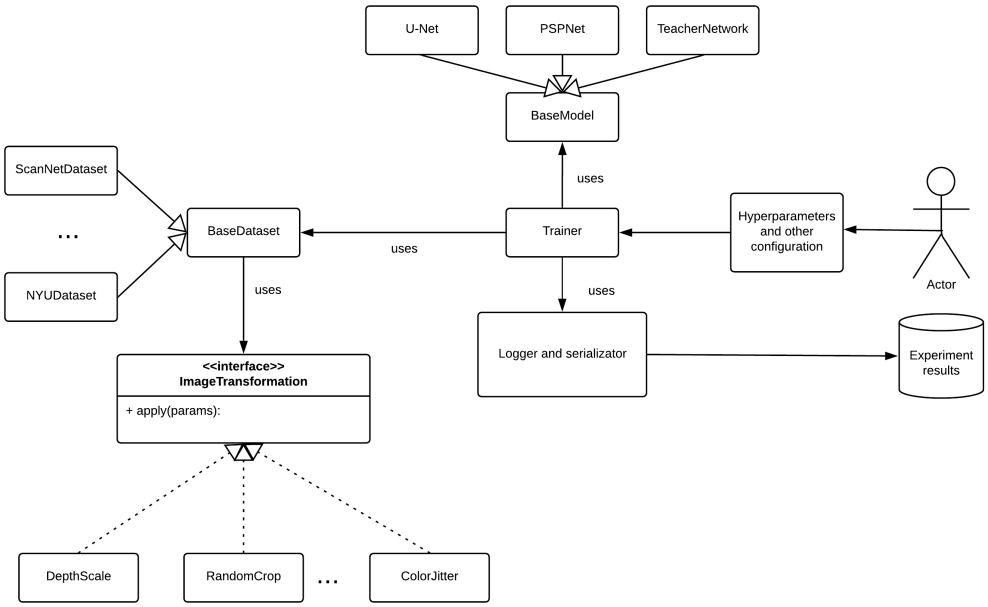



Figure 4.3: Software architecture

## 4.5 Experiment analysis

### 4.5.1 Teacher network

Teacher network converges really fast and has no generalization problems as both training and validation estimation loss is quite similar. Although, validation one seems more unstable having peaks for all the experiments as seen in Figure 4.4. At the same figure we observe that the total training loss suddenly increases at step 3k for one experiment, it is caused because we added $L_{grad}$ after 3 epochs of training with only $L_{depth}$ following Haofeng Chen implementation of Feature Pyramid Network[5]. However, this subtle change does not impact the final performance of the network .

As validation images are not shuffled at each epoch the coincidence in position of big errors for different experiments can be explained by specially difficult scenes such as bathrooms which are less represented in the dataset or complex scenes.

The teacher architecture has been compared with a residual auto-encoder to make sure that the network is making use of the RGB information for the reconstruction. This experiment proved that using both RGB and depth information slightly improves the performance of a depth auto-encoder as seen on Figure 4.5.

The best results were obtained using Adam otimizer, a learning rate of $1e{-}4$, weight decay of $4e{-}5$ and training the model for 13 epochs.

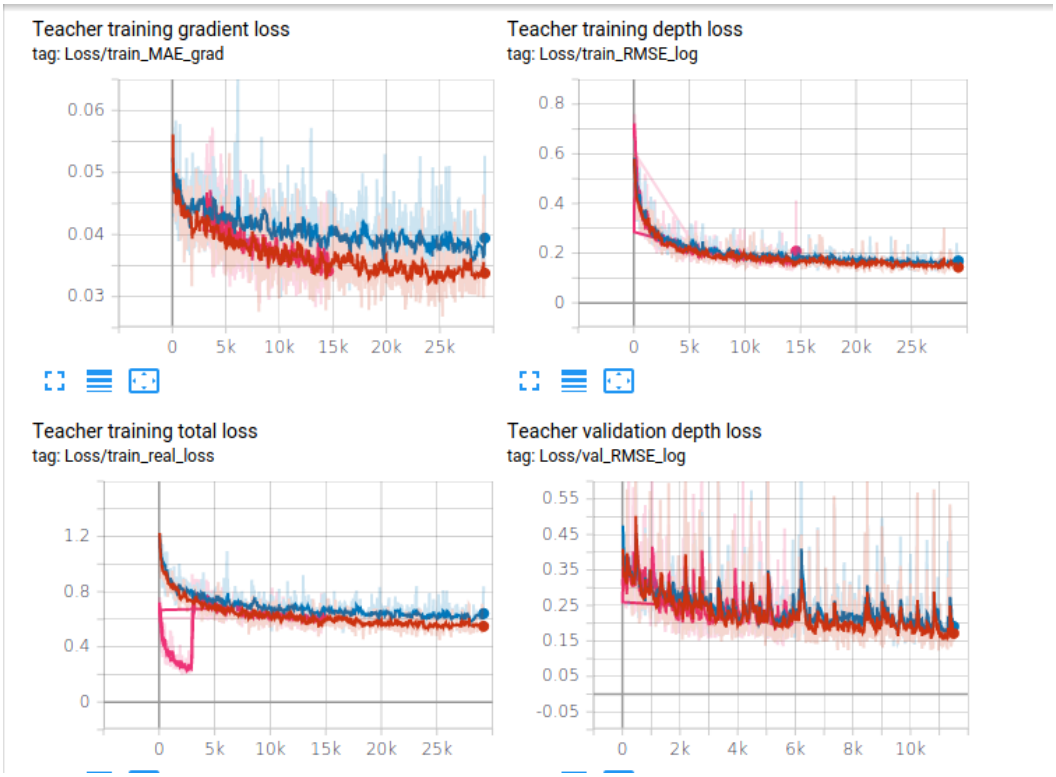



Figure 4.4: Losses from multiple teacher architecture

### 4.5.2 Student network

Once we have a teacher network with similar performance to the state-of-the-art we need to distillate this knowledge into a new network that does not use depth information.

Figure 4.6 shows the loss evolution for 2 different experiments. In the red one both distillation ($L_{emb}$, $L_{repr}$) and reconstruction losses are used during all the training phase. On the other hand, experiment drawn in orange used distillation losses only on the first epochs of the training.

After trying several times to improve baseline U-Net results it has not been possible as seen in Figure 4.7. Those additional constraints added by distillation losses are limiting the learning of our network instead of helping it find a more optimal solution.

### 4.5.3 Using teacher network without privileged information

The results of softly adapting the teacher network to estimate depth maps without privileged information by means of domain adaptation are similar to the ones obtained at previous section. Although teacher networked performed much better than baseline this difference is lost during the domain adaptation phase.

The distance between original depth features and adapted RGB features during training can be seen at Figure 4.8

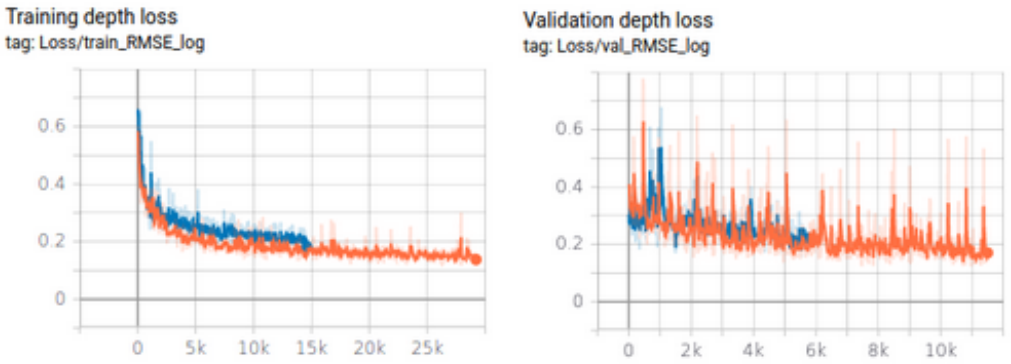



Figure 4.5: Comparison of a residual depth auto-encoder (blue) and our teacher network(orange)

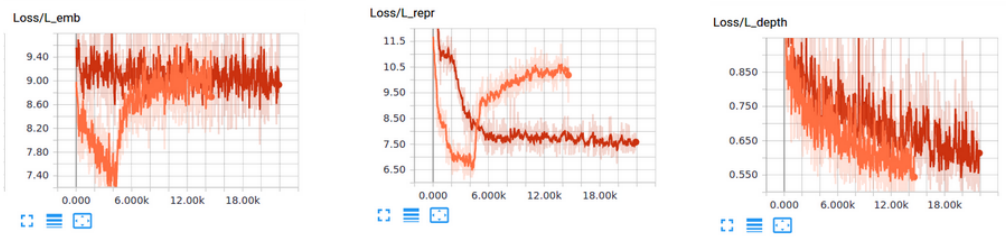



Figure 4.6: Distillation and reconstruction losses of student learning.

## 4.6 Quantitative results

Table 4.2 shows the quantitative results for the most relevant models and architectures developed during this thesis. We can see that it has not been possible to improve U-Net performance using domain adaptation and teacher-student learning.

| | RMSE(lin) | RMSE(log) |
|---|---|---|
| U-Net baseline | 0.988 | 0.440 |
| Teacher network with privileged information | 0.601 | 0.210 |
| Teacher network without privileged information | 1.228 | 0.572 |
| Student network | 1.080 | 0.510 |
| PSPNet baseline | 1.192 | 0.471 |

Table 4.2: Estimation errors for the experiments done in this thesis

If we move forward and compare the results obtained with the latest publications that use the same dataset we will see that our teacher network benchmark is similar to state-of-the-art systems although they are not comparable as our teacher network uses privileged information. The problem is that when distilling this knowledge to student networks our results move far away from the state-of-the-art and do not even improve our baselines as seen at Table 4.3. One possible explanation for this situation is that distillation loss terms are acting as constraints limiting the capacity of our student networks instead of giving hints to find more robust and optimal solutions.

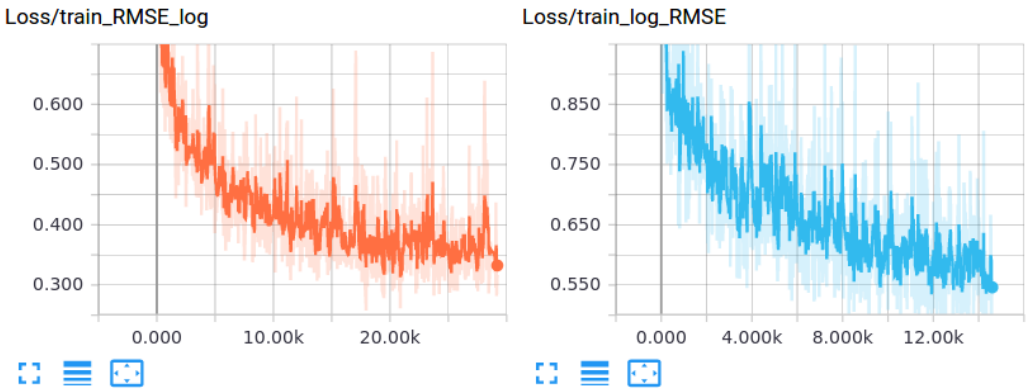



Figure 4.7: Comparison of baseline U-Net $L_{depth}$ compared to our student

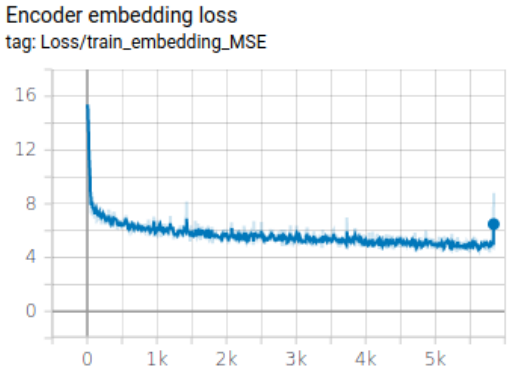



Figure 4.8: Euclidean distance between distilled encoder and teacher encoder features

## 4.7 Qualitative results

Most of the models developed during this thesis struggle to perform on complex scenes while doing good on simpler ones. Furthermore, some shapes confuse our models more than others. One example of this are humans which are rarely found in the dataset. The difference between easy and difficult scenes can be seen at Figure 4.9.

Analysing the predictions at figure Figure 4.10 we observe that a lot of details are lost when transferring our teacher's knowledge to student network, specially the ones that are far away from the camera.

| | RMSE(lin) | RMSE(log) |
|---|---|---|
| Zoran *et al*[28] | 1.200 | 0.420 |
| Liu *et al*[18] | 1.080 | - |
| Baig *et al*[2] | 0.802 | - |
| Laina *et al*[16] | 0.584 | 0.198 |
| Lee *et al*[17] | 0.572 | 0.193 |
| Fu *et al*[10] | 0.547 | 0.188 |
| Teacher network with privileged information | 0.601 | 0.210 |
| Teacher network without privileged information | 1.228 | 0.572 |
| Student network | 1.080 | 0.510 |
| U-Net baseline | 0.988 | 0.440 |

Table 4.3: Quantitative results

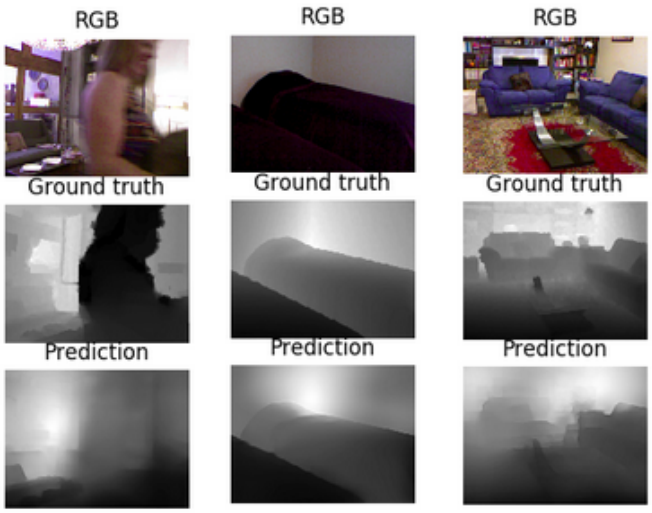



Figure 4.9: Complex scenes at left and right, simple scene in the middle

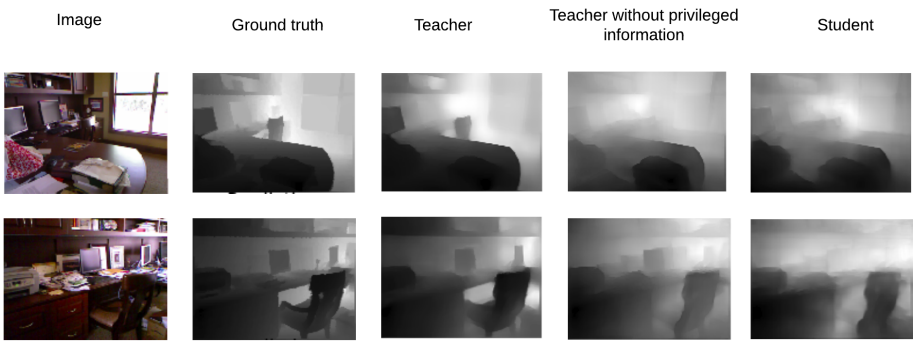



Figure 4.10: Visual comparison of input image, ground truth and relevant estimations

# Chapter 5

# Budget

This thesis has been developed without buying/renting any hardware or software and most relevant costs correspond to computing resources and both student and supervisors wages.

If the project was developed using cloud computing using similar hardware as GPI cluster the minimal instance required would be:

- **CPU cores**: 8
- **GPU Mem**: 8 GB
- **Disk**: 600 GB
- **RAM Mem**: 16 GB

The most similar virtual machine offered at Amazon Web services is the EC2 p2.xlarge instance which costs 0.90 euros per hour. As we will need persistent disk for the dataset we have to add around 12 dollars per month for 600 GB HDD disk.

| | **Amount** | **Wage/hour** | **Dedication** | **Total** |
|---|---|---|---|---|
| Engineer | 1 | 30,00 €/h | 25 h/week | 12,000 € |
| Project supervisors | 2 | 100,00 €/h | 2 h/week | 9,600 € |
| Computation | 1 | 0.9 €/h | 200h | 180 € |
| Storage | 1 | 12 €/month | 4 months | 48 € |
| | | | **Total** | 21,828 € |

Table 5.1: Budget of the project

# Chapter 6

# Conclusions

During the development of this thesis we have successfully implemented a depth estimation system with reasonable performance. Furthermore, we have been able to apply domain adaptation and teacher-student learning even though it has not improved our baseline results.

This has not been an easy journey because of the inner difficulties of debugging deep neural networks and handling big amounts of data. Furthermore, subtle implementation details of the network design and training methodology can provoke significant performance variations.

Although we have not achieved improvements compared to baseline performance applying the teacher-student paradigm several other contributions have been done:

- Training end-to-end depth estimation systems from monocular images.
- Applying teacher-student paradigm to depth estimation.
- Implementing two well known deep learning architectures and comparing them on a challenging task.

As a future work, we want to improve the results obtained with the teacher-student paradigm but also face the problem from other perspectives such as adversarial learning. Furthermore it would be interesting to benchmark our models with completely different datasets such as KITTI[11] which contains outdoor scenes.

# Bibliography

[1] Amina Asif, Muhammad Dawood, and Fayyaz ul Amir Afsar Minhas. A generalized meta-loss function for regression and classification using privileged information, 2018.

[2] Mohammad Haris Baig and Lorenzo Torresani. Coupled depth learning, 2015.

[3] Shaofeng Cai, Yao Shu, Wei Wang, Meihui Zhang, Gang Chen, and Beng Chin Ooi. Effective and efficient dropout for deep convolutional neural networks, 2019.

[4] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 742–751. Curran Associates, Inc., 2017.

[5] Haofeng Chen. Project title. `https://github.com/xanderchf/MonoDepth-FPN-PyTorch`, 2017.

[6] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation, 2019.

[7] Yuru Chen, Haitao Zhao, and Zhengwei Hu. Attention-based context aggregation network for monocular depth estimation, 2019.

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network, 2014.

[10] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[13] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries, 2018.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2016.

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[16] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks, 2016.

[17] J. Lee and C. Kim. Monocular depth estimation using relative depth maps. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9721–9730, June 2019.

[18] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image, 2014.

[19] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[20] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[22] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christopher Bregler. Efficient object localization using convolutional networks, 2014.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[24] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, Dec 2019.

[25] Chenxi Zhang, Liang Wang, and Ruigang Yang. Semantic segmentation of urban scenes using dense depth maps. In *ECCV*, 2010.

[26] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2016.

[27] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video, 2017.

[28] D. Zoran, P. Isola, D. Krishnan, and W. T. Freeman. Learning ordinal relationships for mid-level vision. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 388–396, Dec 2015.