Multi-modal Auto-Encoders as Joint Estimators for Robotics Scene Understanding

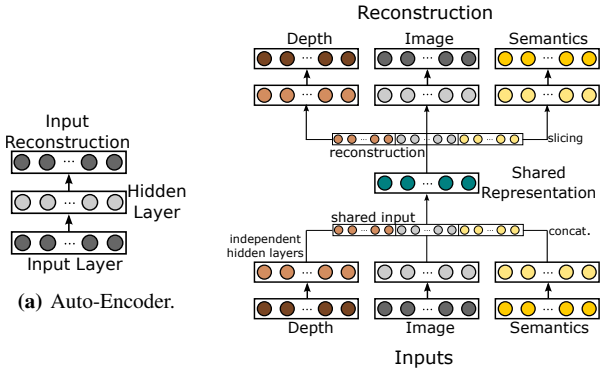
Cesar Cadena
Autonomous Systems Lab
ETH Zurich
Zurich, Switzerland 8092
Email: cesarc@ethz.ch

Anthony Dick and Ian D. Reid
School of Computer Science
The University of Adelaide
Adelaide, Australia 5005
Email: {anthony.dick, ian.reid}@adelaide.edu.au

***Abstract*—We explore the capabilities of Auto-Encoders to fuse the information available from cameras and depth sensors, and to reconstruct missing data, for scene understanding tasks. In particular we consider three input modalities: RGB images; depth images; and semantic label information. We seek to generate complete scene segmentations and depth maps, given images and partial and/or noisy depth and semantic data. We formulate this objective of reconstructing one or more types of scene data using a Multi-modal stacked Auto-Encoder. We show that suitably designed Multi-modal Auto-Encoders can solve the depth estimation and the semantic segmentation problems simultaneously, in the partial or even complete absence of some of the input modalities. We demonstrate our method using the outdoor dataset KITTI that includes LIDAR and stereo cameras. Our results show that as a means to estimate depth from a single image, our method is comparable to the state-of-the-art, and can run in real time (i.e., less than 40ms per frame). But we also show that our method has a significant advantage over other methods in that it can seamlessly use additional data that may be available, such as a sparse point-cloud and/or incomplete coarse semantic labels.**

I. Introduction

In a mobile robotic platform, real-time imagery, scene depth and semantic scene labels potentially provide crucial information for navigating through and interacting with the scene. Within the robotics community the trend has been to rely on expensive sensing suites including, e.g., laser ranging systems, and to perform inference about scene labels using both depth and image data. Recently, there has been significant interest in the computer vision community in the possibility of inferring the depth of the scene from 2D camera images alone, by training a system using large datasets comprising both imagery and depth. However often these algorithms for depth estimation or semantic labelling (or both) are time consuming and therefore not appropriate in a robotic vision context in which real-time constraints are present. Our work is motivated by these recent computer vision successes that apply learning to capture prior information about the relationship between local and global image features and their depth in a scene. However we seek a method that (i) permits real-time inference; and (ii) does not disregard other information that may be available from the sensor/algorithm suite, such as a sparse point cloud data or rough semantic segmentation of the scene, but instead uses it seamlessly to improve the scene estimates.



(a) Auto-Encoder.

(b) Multimodal Stacked Auto-Encoder.

Fig. 1: Learning Models. (a) shows an AE with only one input, its hidden layer and its input's reconstruction as the output. (b) show a MAE after stacking three independent AEs by concatenating their independent hidden layers and learning a shared representation (another AE) to reconstruct the concatenation.

To that end, in this work we explore the capabilities of the denoising Auto-Encoder (AE) (Fig. 1a) [25] to fuse the information available to estimate missing data, even when there exists a complete absence of some of the sensors or modalities. We tackle the problem using a Muti-modal stacked denoising Auto-Encoder (MAE) [20], which handles three input modalities – RGB image data; scene depth; and semantic information; as illustrated in Fig. 1b – and apply this model to the problem of outdoor scene understanding for a mobile robotic platform. A significant advantage of this approach is that it allows us to naturally exploit even partial information to improve our predictions; e.g., when estimating the depth or semantics from a single image we would like to use all the information available, such as the sparse depth from a Structure from Motion (SfM) system, or a foreground/background segmentation algorithm.

We are inspired by [20], which fuses audio and video data for speech classification. This work demonstrated that better, more informative, features are learnt when different modalities are taken into account. Furthermore, it demonstrated that after having a suitable training stage, it is then possible to use

the data from one modality (in the absence of data from the other) to recover the missing data at test time. We leverage this insight to estimate the depth and semantics of a scene jointly, given only the RGB information.

Originally, the AE was developed as an unsupervised technique for feature learning. We focus instead on its use for reconstruction. Although the shared representation could be used as a pre-trained feature for other classification tasks, we do not address that possibility in this paper.

II. Related Work

The problem of geometry estimation from a single image has a long history. The earliest work (e.g., [5]) was concerned only with understanding the geometry of the scene and relied on significant manual intervention to create 3D scenes from a single view. Subsequent to this, various works considered how 3D "pop-ups" or more complex scenes could be reconstructed automatically by combining machine learning of scene labels with perspective inversion [9, 12, 13]. These, and a number of more recent learning approaches (such as [15, 17, 23]), have typically relied on hand-crafted features and are computationally expensive, taking on the order of seconds to minutes per image. Saxena *et al.* [23] use only the appearance information to estimate the depth of the scene. Liu *et al.* [17] first estimate a semantic segmentation to guide a better depth estimation – i.e., they use the semantic information as an input to their inference mechanism. On the other hand, Ladický *et al.* [15] propose to jointly estimate the depth and the semantic segmentation obtaining better depth estimation than by depth inference alone. Both depth and semantics are outputs of their system. Our work is related to both: we exploit semantic information to learn models that are able to jointly estimate depth and semantics at test time, even when only the image information is available. Unlike the previous approaches, our model estimates the depth using any semantic information available at test time, and simultaneously (re-)estimates the full semantic segmentation.

An alternative to the models described above is to take a non-parametric, data-intensive approach to depth estimation, notably [1, 21]. In these approaches patches from an image are matched to a database of patches each of which is labeled with its correct depth. The current patch then takes the depth of closest match in the database. In addition to the burden of the hand-crafted features, these approaches require to keep great number of samples to transfer the correct depth to each patch. Moreover, [21] uses this idea to densify an existing depth map, rather than estimate depth from 2D image data only.

Recently, there has been significant interest in the possibility of using Convolutional Neural Networks (CNN) for the depth estimation problem [6, 7, 18]. Eigen *et al.* [7] formulate the depth estimation as regression problem, using multi-scale CNNs to produce a depth map. Liu *et al.* [18] combine a CNN with a continuous Conditional Random Field that encodes scene smoothness priors. Eigen and Fergus [6] extend their previous work to estimate not only depth, but surface normals and semantic label estimation. Although they use the same network structure for each task, each network is learnt independently, and inference is also independent, meaning they do not capitalise on the synergy between the different modalities. In contrast our approach aims explicitly to take advantage of the correlations that exist between the scene's semantic labels and its depth.

These previous deep learning approaches have all been based on CNNs. We deliberately adopt a different architecture since we seek a model that is flexible enough to support partial knowledge from different inputs and powerful enough to be able to estimate the missing parts. For this purpose we make use of the AE learning structure. AEs have been used for related purposes in the past, such as for image denoising and inpainting [26], joint feature learning for speech recognition [20], while [24] explored multimodal learning deep boltzmann machines for image classification using images and text.

III. Auto-Encoders

Auto-Encoders belong to the family of unsupervised neural network models. They are trained to compute a representation of the input (this representation is known as the "code"), from which we can recover the input as accurately as possible [25]. In their most abstract form an AE *encodes* a visible input $\mathbf{v} \in \mathcal{R}^n$, to a hidden representation $\mathbf{h} \in \mathcal{R}^m$ through a deterministic mapping:

$$\mathbf{h} = \sigma_e(\mathbf{W}_e\mathbf{v} + \mathbf{b}_e) \tag{1}$$

where σ is a non-linear function. The hidden representation $\mathbf{h}$ – i.e., the code – is then *decoded* into a reconstruction $\hat{\mathbf{v}}$ with the same dimensions as $\mathbf{v}$, (see Fig. 1a) through:

$$\hat{\mathbf{v}} = \sigma_d(\mathbf{W}_d\mathbf{h} + \mathbf{b}_d) \tag{2}$$

$\hat{\mathbf{v}}$ should be seen as a prediction of $\mathbf{v}$, given the code $\mathbf{h}$. The parameters $\mathbf{W}_e$, $\mathbf{b}_e$, $\mathbf{W}_d$ and $\mathbf{b}_d$ are optimized during training such that the average reconstruction error on the training set is minimized.

Since the objective of an AE is to reconstruct the signal from the hidden layers, the typical loss function used for training considers the reconstruction error. This can be measured in many ways depending on the assumed distribution of the input given the code. Often the squared error $L(\mathbf{v}, \hat{\mathbf{v}}) = ||\mathbf{v} - \hat{\mathbf{v}}||_2^2$, is used. We follow this convention in our work.

If the dimensionality of the hidden layer is greater than or equal to the input layer a trivial (identity) solution could be learnt. Therefore, different strategies have been proposed to learn useful representations in the hidden layer [2, 3, 25]. One popular strategy is the *denoising* AE [25]. In a denoising AE the input data is randomly corrupted during training; since we want to recover the original un-corrupted input, this deliberate corruption forces the hidden representation to learn a more global structure of the input.

This ability of AEs to *clean* the inputs under missing data (e.g., missing parts of a depth map) can be extended to recover full chunks of missing data (e.g., missing an entire depth

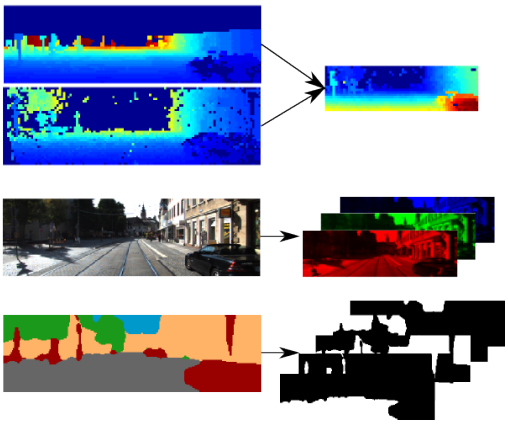


Fig. 2: Inputs during training. Top: we enrich the sparse depth data coming from the LIDAR with the sparse depth coming from a stereo computation [10], then it is parametrized as the inverse depth. Middle: the image is processed independently for each channel until a shared representation is needed. Bottom: a semantic segmentation over the image is obtained [16] and then we independently process each class as binary image until a shared representation is learned; semantic color code: *ground*, *objects*, *building*, *vegetation* and *sky*.

map). In other words, while we train with full information (i.e., all sensor modalities), the model can recover some of those modalities if they are missing at test time. It is this *opportunistic* use of *whatever* data that is present at test time that makes AEs so attractive for deployment in our work. We explore this in more detail in the next section.

IV. Proposed Architecture

In this section we describe our MAE learning architecture along with other possible network topologies to estimate the same missing information (depth and semantics).

One (naive) approach is to simply concatenate the different input channels and try to learn a single AE. However, [20] demonstrated that this option was not able to learn useful representations of the intrinsic correlations across modalities. Their work showed that it was better first to learn useful independent representations and then learn the correlations among those features. Their result motivates our approach as follows.

The first step in our MAE model is to learn an independent denoising AE for each input modality, as in Fig. 1a. The input modalities that we handle are: RGB images; sparse depth images (such as those provided by LIDAR); and the coarse semantic classes ground, building, vegetation, sky and "other", a general object class (see Fig. 2). This selection give us a total of nine channels, three for RGB, one for depth and five for semantics, in which each semantic class input is a binary mask.

A. Models

It is possible to learn the shared representations that capture the inter-relationships among the inputs using networks with different topologies. In this paper we explore three of them, as shown in Fig. 3(a,c,d).

a) Full Flat MAE:: (*flat-MAE*) In this model we directly concatenate the hidden layer for each channel in a full stacked AE, similarly to [20] (Fig. 3a). We aim to learn a full shared representation capturing the inter-relationships or correlations across all inputs.

b) Full MAE:: (*full-MAE*) In this model we first stack the AEs corresponding to the semantic classes to learn a shared representation that will capture the global context among these coarse classes, Fig. 3b. This semantic shared representation is then concatenated with the hidden layers for the depth and RGB channels in a full stacked AE, see Fig. 3c.

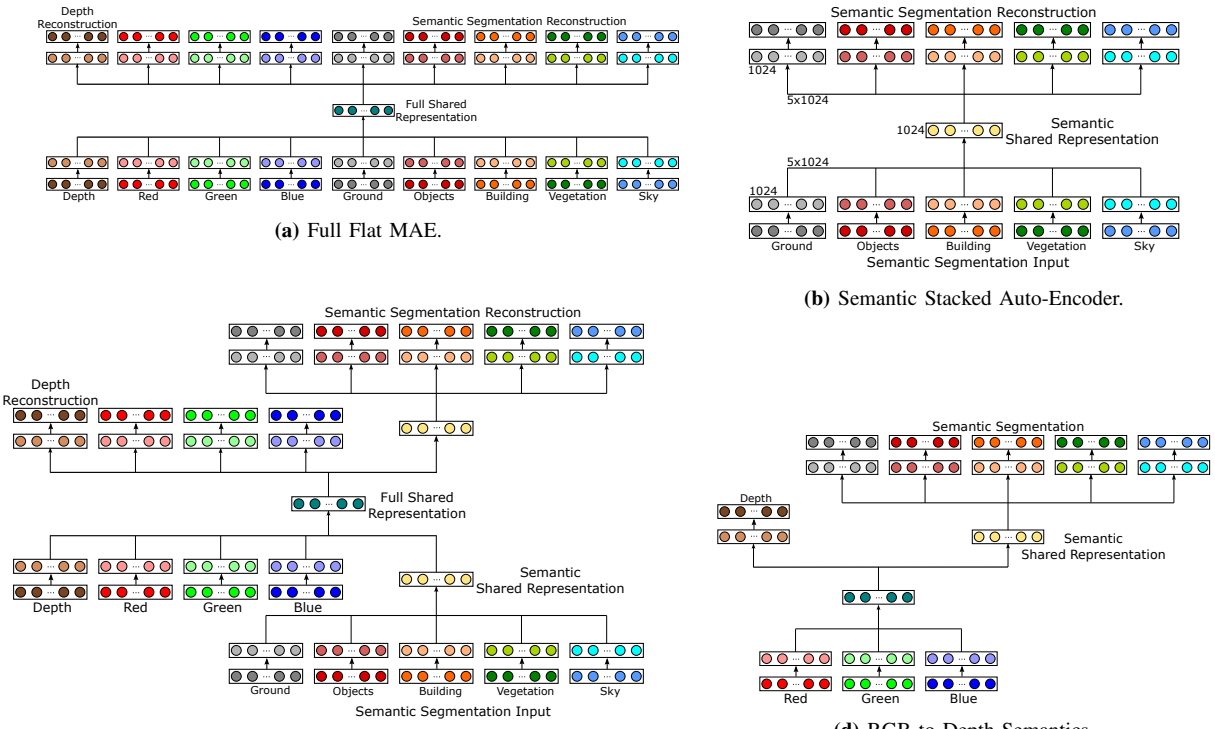
c) RGB to Depth-Semantics:: (*rgb2sd*) Here we use the encoders from the RGB stacked with the decoders from the depth and semantic AEs, see Fig. 3d. This model is the closest one to a standard supervised learning model that tries to predict depth and semantics from images. As with the other models we use the AE pre-training stage to obtain initial estimates for the network parameters, and then use images as inputs and the corresponding known depth and semantics as output "labels". We use this model as a baseline to illustrate the benefits of models *flat-MAE* and *full-MAE* which learn a shared hidden representation.

B. Training setup

For each independent denoising AE we corrupt the input data by forcing 10% of pixels to be zero. The only data augmentation that we have used is horizontal flip for all the examples. For RGB and semantic channels we use the Rectified Linear Unit (ReLU) activation function in the encoder, Eq. 1, and the sigmoid activation functions in the decoder, Eq. 2. For reconstruction we choose the Euclidean loss function on these channels. Note that each semantic class is a separate input, coded as a binary image mask; for these data the Euclidean loss on the sigmoid function is approximately a zero-one loss.

We parametrize the 3D information as the *inverse depth* in the depth channel, which allows for representation of points that are effectively at infinite depth (e.g., sky), and has better convergence properties when estimating the 3D. In this channel we use mean subtraction, ReLU in the encoder, and no activation function in the decoder. Active depth sensors have different blind spots, for instance due to specularity or out of range measures, making the depth input sparse in most cases. For this reason we use a Euclidean loss over only the valid depth data. For the non-valid depth we assign the loss, and its gradient, to zero.

When stacking the AEs we use ReLUs for both encoder and decoder parts in the shared representations. Since each stacked AE is composed of smaller components that are already trained, we copy the parameters from these to pre-train the stacked versions. New parameters in a stacked model, which are not inherited from a smaller pre-trained model, are randomly initialized following a zero-mean Gaussian. We first run an initialization training stage for a few epochs, allowing only the new parameters to be updated, while the pre-trained



(a) Full Flat MAE.

(b) Semantic Stacked Auto-Encoder.

(c) Full MAE.

(d) RGB to Depth-Semantics.

Fig. 3: Auto-Encoder models to estimate depth and semantic segmentation.

ones are kept fixed. After the initialization epochs, we then optimize all the parameters.

Since we aim to reconstruct the full scene in the presence of missing data (most ambitiously to estimate the depth and semantic segmentation from a single monocular image), it is essential to augment the training data with examples of missing data. For example, as well as training the MAE with a full set of RGB, semantics, and depth (at both its input and output), we also train it with RGB alone presented at the input (and the full set at the output). Of course this applies only to *full-MAE* and *flat-MAE*, and not to *rgb2sd*.

The size of the models, pre-training setups, and training parameters are illustrated in Table I. There, *ind.* refers to the hidden layer of each channel's AE.

V. Experiments

We have evaluated our model on outdoor scenes from the KITTI dataset [11]. This dataset provides stereo RGB images and 3D point clouds from a rotating LIDAR scanner. We use the same sequences for training and testing as proposed by [7].

We use both images from the each stereo pair as independent examples, and project the 3D information on each one to obtain the ground truth depth using the official toolbox provided with the dataset. This projection results in sparse depth images covering the bottom part of the RGB images. We enrich the depth evidence in the upper part of the RGB images by computing disparity (i.e., inverse depth) from the stereo pair. This has the advantage that we include evidence that is otherwise missed by the LIDAR such as the tops of trees and buildings.

The semantic segmentation ground truth is not available for these sequences. Thus, we determine an "approximate ground truth" for all images using a top performing system [16][1]. We fine-tuned [16] using the semantic segmentation hand labeled data (140 images), a subset of the same KITTI dataset, made available by [28]. Although computationally expensive ($> 2s$ per image), [16] provides highly accurate segmentations on the testing set (112 images), with an average accuracy of 94.5% per class (96.4% per pixel accuracy). By determining a proxy ground truth in this manner we run a small risk of training our system to make the same errors as [16]; this expedient does not diminish our overall contribution and it would be trivial for us to retrain using actual ground truth if such were available.

The original RGB images are around 1240x370 pixels in size, but each sequence has different image size. As a starting point for comparisons between different AE models we down-sampled them to 60x18 pixels. Later, the *full-MAE* is retrained to handle 120x36 and 240x72 down-sampled versions.

The training set has 30602 images in total, augmented with a horizontally flipped version in each case, resulting in a dataset of 61204 training images. We used all the images (and depth maps) in each sequence even when the car has stopped. This is because even when the scene remains the same, the sensor

[1]at the time of writing, the top performer on PASCAL VOC 2012 segmentation

TABLE I: Models settings.

Model	pre-training	layer size				learning rate (at epoch)	total epochs	params. [M]	time training
		input	ind.	sem.	full				
60x18 resolution									
independent-AEs	—	1080	1024	—	—	1e-2, 1e-3(100)	150	2.2	10min
Semantic-MAE	S-AEs	5x1080	5x1024	1024	—	1e-3, 1e-4(100)	150	21.6	45min
full-MAE	D,R,G,B-AEs and S-MAE	9x1080	9x1024	1024	1024	1e-3, 1e-4(100)	150	41.9	1h40min
flat-MAE	D,R,G,B-S-AEs	9x1080	9x1024	—	1024	1e-3, 1e-4(100)	150	38.8	1h10min
RGB to Depth-Sem.	D,R,G,B-AEs and S-MAE	1080	1024	1024	1024	1e-3, 1e-4(100)	150	20.5	1h
120x36 resolution									
full-MAE	full-MAE 60x18	9x4320	9x1024	1024	1024	1e-5, 1e-6(20), 1e-7(50)	60	101.7	2h
240x72 resolution									
full-MAE	full-MAE 120x36	9x17280	9x1024	1024	1024	1e-6, 1e-7(20), 1e-8(50)	60	340.7	18h

measurements are different.

A. Depth estimation results

We evaluate the depth estimation with different error metrics that have been proposed in previous works [7, 22]. There are 697 frames from 28 different sequences in the testing set. When evaluating, all the depth predictions are up-scaled by bilinear interpolation to match the corresponding frame size of each sequence.

With d_p and $\hat{d}_p$ denoting the ground-truth and predicted depths respectively at pixel p, and T the total number of pixels, with valid ground truth and prediction, in all the evaluated depths, the metrics we use are: the Absolute Relative Error, $\frac{1}{T}\sum_p \frac{|d_p - \hat{d}_p|}{d_p}$; the linear RMSE, $\sqrt{\frac{1}{T}\sum_p (d_p - \hat{d}_p)^2}$; the log scale invariant RMSE, $\frac{1}{T}\sum_p (\log \hat{d}_p - \log d_p + \alpha(\hat{d}_p, d_p))^2$; and the Accuracy under a threshold, $\max\left(\frac{\hat{d}_p}{d_p}, \frac{\hat{d}_p}{d_p}\right) = \delta < th$.

In our first experiment we compare different learning models to estimate the depth information at resolution of 60x18 pixels, where the depth input is set to zeros. We show the errors for several models computed on the full testing dataset in Table II.

In addition to the models described in Section IV-A we have trained the multi-modal AE without the semantic information (rgbd-MAE) and a model with only the rgb encoder and the depth decoder (rgb2d). In general, better results are obtained by the multi-modal AEs, with the *full-MAE* model providing the best performance. As reference, we also include the results of a standard stereo matching system.

We detail in Table II the information used as an input for each evaluation, the color image of the scene (R,G,B), the semantic segmentation (S) and the sparse depth on extracted FAST corner keypoints (sD). In red we highlight the best values using only the color image as the input. When using the semantic information as well the best model is the *full-MAE* model at 124x72 resolution, (number highlighted in blue). The best metrics in the full comparison (excluding the stereo) are in bold.

Let's take a closer view of Table II. When comparing the models at 60x18 with RGB-input only, the model *rgb2d* has the lowest errors while our *full-MAE* is the second best on errors and obtains the highest accuracies on 2 out of 3 thresholds.

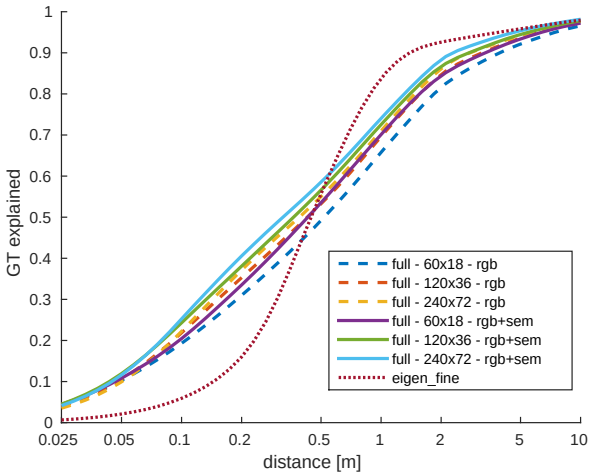


Fig. 4: Ratio of the explained 3D point cloud ground truth by the 3D point cloud from the depth estimation of different models (y-axis) up to some maximum distance (x-axis). We show the state-of-the-art model (eigen_fine [7]) on KITTI dataset, and our model in three resolutions and using only RGB or RGB plus semantics.

If we assess the effect of using more information in MAEs, a consistent improvement at each resolution is clear by using either the semantic segmentation or the sparse depth. Furthermore, the best results for each metric are obtained when we use both semantics and sparse depth.

It is not surprising that the performance improves alongside the resolution of the MAE models, as they have access to more information. However we also observe that the impact of using sparse depth is not as great at the 240x72 resolution than at lower resolutions. We believe this is because each keypoint in the higher resolution has to influence a larger number of pixels, which could result in greater difficulty for the network to learn the features.

We up-scale our *full-MAE* model to handle 120x36 and 240x72 pixels resolution inputs. We initialize the weights

TABLE II: Comparison of depth estimation on the KITTI dataset. The inputs column is coded as R,G,B, for the image color channels, S for the 5-channel semantic inputs, and sD for sparse depth from the stereo matching on a corner detector.

Method			Errors (lower is better)			Accuracy (higher is better)		
Model	Inputs	out.res.	abs.rel.	rms [m]	log.sc.inv	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
stereo			0.077	4.36	0.179	0.939	0.969	0.982
eigen_fine [7]	R,G,B	144x27	0.320	8.08	0.509	0.512	0.822	0.922
rgb2d	R,G,B	60x18	0.275	9.11	0.360	0.537	0.758	0.879
rgb2sd	R,G,B	60x18	0.290	9.18	0.363	0.530	0.756	0.879
rgbd-MAE	R,G,B	60x18	0.300	9.34	0.368	0.527	0.753	0.873
flat-MAE	R,G,B	60x18	0.290	9.50	0.368	0.536	0.757	0.872
flat-MAE	R,G,B,S	60x18	0.255	8.87	0.335	0.588	0.796	0.897
full-MAE	R,G,B	60x18	0.288	9.44	0.367	0.540	0.761	0.875
full-MAE	R,G,B,S	60x18	0.252	8.76	0.327	0.586	0.802	0.903
full-MAE	R,G,B,sD	60x18	0.199	7.78	0.290	0.654	0.859	0.943
full-MAE	R,G,B,S,sD	60x18	0.184	7.52	**0.276**	0.687	0.877	0.948
full-MAE	R,G,B	120x36	0.286	8.99	0.371	0.578	0.781	0.887
full-MAE	R,G,B,S	120x36	0.250	8.34	0.338	0.617	0.815	0.909
full-MAE	R,G,B,sD	120x36	0.202	7.53	0.311	0.671	0.870	0.947
full-MAE	R,G,B,S,sD	120x36	**0.179**	7.14	0.297	**0.709**	**0.888**	**0.956**
full-MAE	R,G,B	240x72	0.291	8.65	0.363	0.597	0.791	0.894
full-MAE	R,G,B,S	240x72	0.243	7.80	0.323	0.643	0.833	0.925
full-MAE	R,G,B,sD	240x72	0.220	7.61	0.317	0.660	0.856	0.940
full-MAE	R,G,B,S,sD	240x72	0.194	**7.10**	0.295	0.695	0.881	0.954

for these models with the ones from the immediate lower resolution in the internal layers, and properly up-scale the size of the matrices $\mathbf{W}$s and $\mathbf{b}$s affected by the change in size[2]. Note that our up-scaled decoders still make use of the same sized hidden layer. It is possible that increasing this size would permit the higher resolution decoders access to more detailed information than is available at present, but we defer this experiment for future work.

In row 2 of Table II we also show the performance of the state-of-the-art depth estimation methods, the model proposed in [7] (depth predictions downloaded from the authors' website) *eigen_fine* in Table II, which has 90.9M parameters in the model. Their model outputs are 144x27 pixels, corresponding to a partial coverage of the image and ground truth depth.

The single metrics of Table II do not tell the full story for a meaningful comparison. For example, the ordering of performance permutes depending on the chosen metric and these metrics do not reflect the sparsity or the coverage of the ground truth and the predictions. For these reasons we report the ratio of the ground truth explained by the predictions for different maximum distances in Fig. 4. The curves are built by projecting the predicted depths to a 3D point cloud and computing the distances for each 3D point in the ground truth to the closest predicted 3D point, then computing the ratio of the distances that are less than a certain threshold.

As shown in Fig. 4, the amount of explained ground truth increases with the resolution of the models for any selected maximum distance. The estimations from [6] perform better than our model only when more than 50cm of error are allowed.

Note that a direct comparison on a level playing field is almost impossible and we provide these comparisons for reference only. There are various factors involved in how the two systems are trained and these should be borne in mind when interpreting the data. For example: [7] executes more comprehensive data augmentation to have a training set size of 1.5M samples; though our training set is smaller, we allow the system access to the semantic information at training time to guide the learning of the shared representation.

[2]We explored different initialization options: random, learning again for each stage, and RBMs, and found the up-scaling initialization gives the best results for the same number of epochs.

B. Semantic Segmentation Results

In the previous section we evaluated the depth reconstruction obtained with our *full-MAE* using different inputs. In this section we evaluate the semantic segmentation output when only the RGB image is available as an input; that is, the depth and semantic inputs are both set to zeros.

In order to show how our *full-MAE* semantic segmentation performs we evaluate it on the same hand-labeled set of [28]. We have selected the images from [28] that do not overlap with any of the sequences in our training, resulting in 140 images. We report the recall accuracy and the intersection over union of our *full-MAE* semantic segmentation in Table III. We also report the results of *rgb2sd* and *flat-MAE*. These results demonstrate that, even under the same inputs, learning the synergies among the different modalities with the multi-modal AEs leads to better estimations for all the classes.

Finally, our *full-MAE* obtains the best overall performance, meaning that our architecture, Fig. 3c, learns better correlations among the semantic classes than a flat model.

For the sake of completeness we also include the results from two semantic segmentation systems, [28] and [4], in Table III. Please note that a direct comparison is not possible given that the test sets are different (marked as ‡). Although [28] performs better on the semantic segmentation task, the computation cost for their approach is of the order of several

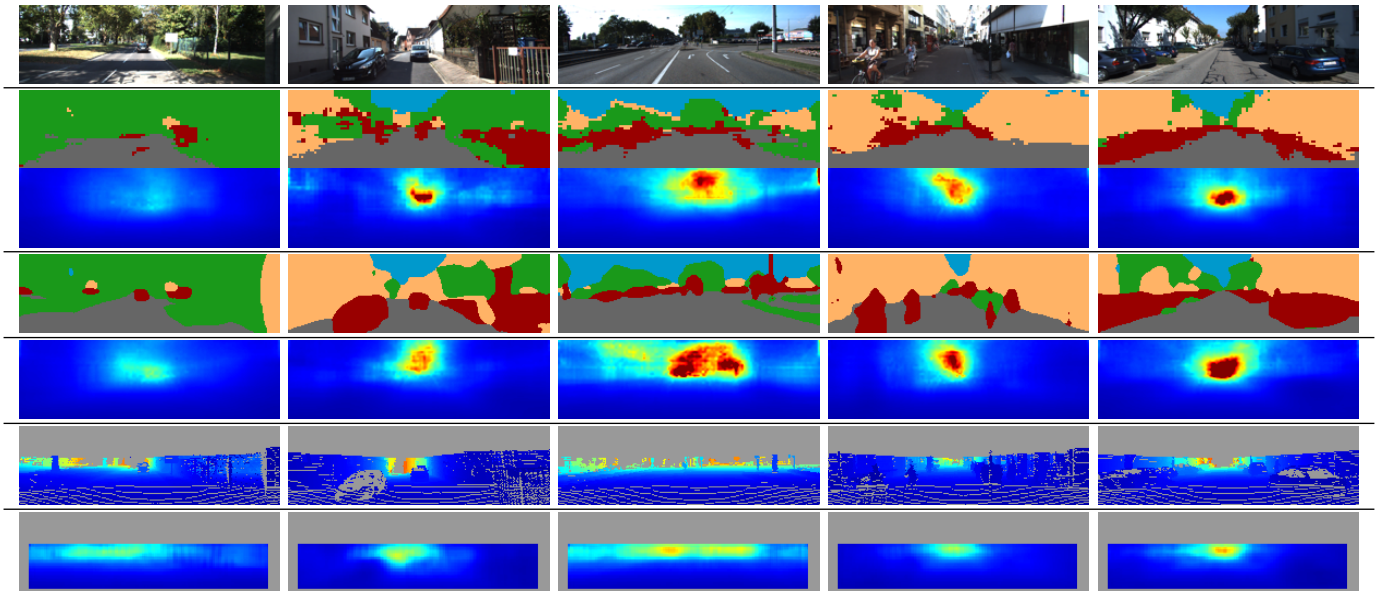


Fig. 5: Reconstruction results from our *full-MAE* model. The first row shows the RGB input to the model. In the second and third row we show the reconstruction for semantics and depth by setting the corresponding inputs to zero. The fourth row shows an alternative image-only semantic segmentation [16] used as the semantic input, and the depth reconstruction result of using this together with RGB is shown in the fifth row. The last two rows correspond to the ground truth depth and the results from [7].

seconds per frame, while our approach provides the semantic segmentation and the dense depth estimation in a matter of milliseconds per frame, making it potentially much more suitable for robotics applications. For a more direct quantitative comparison, we retrain [4] and evaluate it on the same test set than ours. Our *full-MAE* running faster than 25fps outperforms [4] which runs at 5fps.

TABLE III: Semantic segmentation evaluation with image-only inputs. Note that the results for [28] are taken directly from Table II in that paper. Their results refer to a different test set (‡), and require orders of magnitude more computational time than ours. We present them for the sake of completeness.

Accuracy [%]

Model	bldg	sky	gnd	veg	obj	avg.	pixel
[4]-Im ‡	93.2	91.9	78.6	93.2	32.5	76.9	83.2
[28]-Im ‡	87.5	92.5	91.9	92.5	66.1	86.1	89.4
[4]-Im	**92.7**	83.9	80.4	**86.7**	18.3	72.4	77.9
rgb2sd	62.2	94.1	94.1	82.2	33.8	73.3	76.9
flat-MAE	66.0	95.5	94.5	86.5	35.1	75.5	79.4
full-MAE	69.0	**96.8**	**95.8**	86.4	**42.6**	**78.1**	**81.3**

Intersection over Union[%]

Model	bldg	sky	gnd	veg	obj	average
[4]-Im	55.5	82.3	71.8	**77.3**	17.3	60.8
rgb2sd	52.5	79.0	78.3	62.8	27.8	60.1
flat-MAE	55.2	80.8	81.7	67.4	30.6	63.0
full-MAE	**59.2**	**83.1**	**82.4**	69.6	**36.5**	**66.2**

C. With Partial Inputs

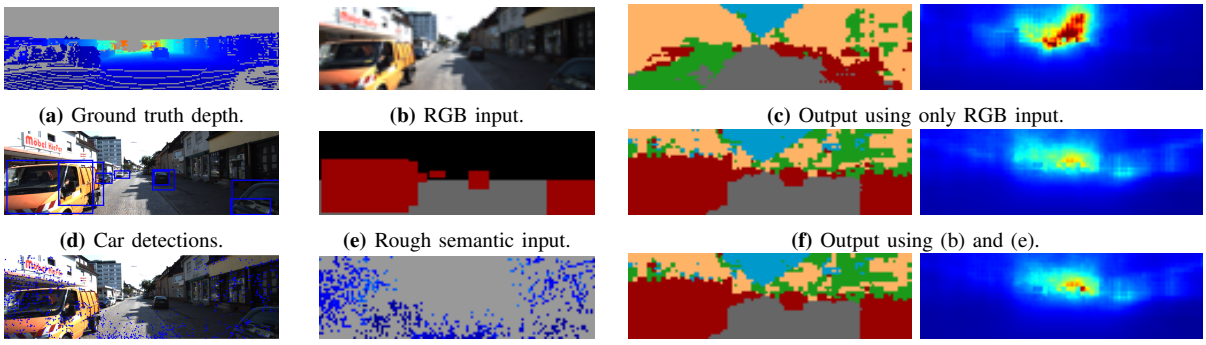
A major advantage of our method is its ability to use additional information available to obtain a better depth and semantic segmentation. To illustrate this, we take one complete sequence of those in the testing set of [7], and first estimate the depth and semantics using only the RGB input. An example frame is shown in Fig. 6c.

TABLE IV: Depth accuracy with partial and noisy inputs.

Errors	Inputs: RGB	RGB + P.Sem	RGB + SfM	RGB + P.Sem + SfM
abs. rel.	0.304	0.270	0.292	**0.251**
RMSE(linear) [m]	7.94	7.01	7.85	**6.96**
RMSE(log.sc.inv.)	0.359	0.332	0.350	**0.321**
Accuracy				
$\delta < 1.25$	0.429	**0.614**	0.442	0.610
$\delta < 1.25^2$	0.695	0.837	0.714	**0.838**
$\delta < 1.25^3$	0.887	0.929	0.896	**0.930**

Next, to show how even partial semantic information is beneficial, we run a standard object detector [8] on this sequence to detect cars, Fig. 6d. With the bounding boxes for cars declared as *objects* class and one mask for the *ground* class on the bottom of the image, we have a rough semantic segmentation input, Fig. 6e. With this rough and incomplete segmentation and the RGB as inputs, our model estimates the dense depth and full semantics as shown in Fig. 6f.

As a final step, we compute a sparse point cloud reconstruction for this sequence using the SfM implementation, a monocular visual odometry system, of [10], see Fig. 6g. We now use the network to re-estimate using only the RGB and this noisy and sparse depth, as well as using all the inputs available so far: RGB, sparse depth and partial semantic segmentation. Fig. 6i shows the output.



(a) Ground truth depth. **(b)** RGB input. **(c)** Output using only RGB input.

(d) Car detections. **(e)** Rough semantic input. **(f)** Output using (b) and (e).

(g) Point cloud from SfM. **(h)** Sparse depth from SfM. **(i)** Output using (b), (e) and (h).

Fig. 6: Qualitative results using partial information as inputs.

We report a quantitative comparison on the depth estimation under these four scenarios in Table IV. It is clear how every piece of extra information can be seamlessly incorporated in our *full-MAE* to improve the estimation. One interesting aspect of this table is that the addition of the "sparse semantics" appears to be of greater assistance than the sparse depth. We speculate that this is because the system is most likely to make depth errors around objects such as cars, and the addition of the semantic information helps to prevent these errors, having a greater impact on the overall depth performance than the noisy sparse depth.

D. Computational and Memory Requirements

We used *caffe* [14] to implement, train and test all the learning models. The training was carried out using an nVidia GeForce GTX TITAN X GPU with 3072 cores and 12 Gb memory. The training times are reported in the last column of Table I.

All the evaluation was performed using an nVidia GeForce GTX-680 GPU with 1536 cores and 2Gb memory. Processing the testing set for our *full-MAE* model with 60x18 resolution took 11.2ms for batches of 100 frames to estimate the depth and the semantic segmentation. Processing only one frame at a time incurs GPU communication overheads and takes 7.9ms on average (including the overheads). In the *full-MAE* model at 120x36 resolution the timing for batches of 100 was 20ms. Processing only one frame at a time at this resolution takes 12.8ms. With our *full-MAE* model at 240x72 resolution the timing for processing only one frame at a time is 35.5ms.

A more efficient computation is possible when the depth and/or semantic inputs are set to zero, since in this case it is possible to pre-compute the corresponding hidden layers in the encoder stages, saving memory and computational time.

VI. Discussion and Conclusions

We have presented a MAE model for depth and semantic segmentation estimation. By exploiting different modalities and learning a shared representation, our model performs better on tasks such as depth estimation from a single image than a comparable non-shared representation. Even when using imperfect semantic segmentation during training, the MAE model is able to learn useful shared codes between the different modalities, and gives better depth estimations when the semantic knowledge is available. Furthermore our MAE model comes with the benefit of being able to use any available knowledge, even partial data, to arrive at its scene estimation. By way of example, we have shown how our method can densify a depth map using sparse point-cloud data along with a single RGB. To our knowledge no other deep network, such as CNNs or even other learning approaches (e.g., holistic methods [27]), are able to handle missing information – even in extreme cases with full absence of one input modality – to perform the inference.

In a quantitative comparison with a state-of-the-art depth estimation system, our MAE model behaves comparably: the results show either slightly better or slightly worse performance depending on the metric. Our system obtains more accurate estimation for the close range than for long range (a characteristic of the inverse depth parametrization) and this means we perform better on log-scale metrics than linear ones. An interesting question for future work will be to develop a depth loss function that can produce similar accuracy across the full range of useful depths. We also hope to investigate Convolutional Auto-encoders [19] as a means to exploit greater depth of the network (and therefore potentially better inference ability) while retaining the benefits we have demonstrated.

Acknowledgments

We are extremely grateful to the Australian Research Council for funding this research through project DP130104413, the ARC Centre of Excellence for Robotic Vision CE140100016, and through a Laureate Fellowship FL130100102 to IDR. This work was carried out while CC was at the University of Adelaide.

References

[1] M.H. Baig, V. Jagadeesh, R. Piramuthu, A. Bhardwaj, W. Di, and N. Sundaresan. Im2depth: Scalable exemplar based depth transfer. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 145–152, 2014.

[2] Y. Bengio. Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Machine Learning, Volume 7*, page 19, 2012.

[3] Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.

[4] C. Cadena and J. Košecká. Semantic segmentation with heterogeneous sensor coverages. In *Proc. IEEE Int. Conf. Robotics and Automation*, Hong Kong, China, June 2014.

[5] A. Criminisi, I.D. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40 (2):123–148, 2000.

[6] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. Int. Conf. Computer Vision*, 2015.

[7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems 27*, 2014.

[8] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32:1627–1645, 2010.

[9] A. Flint, D. Murray, and I.D. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2228–2235, 2011.

[10] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] D. Hoiem, A.A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics (TOG)*, 24(3): 577–584, 2005.

[13] D. Hoiem, A.A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision*, 80(1):3–15, 2008. ISSN 0920-5691. doi: 10.1007/s11263-008-0137-5.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[15] L. Ladický, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 89–96, 2014.

[16] G. Lin, C. Shen, I.D. Reid, and A. van den Hengel. Efficient piecewise training of deep structured models for semantic segmentation. *CoRR*, abs/1504.01013, 2015.

[17] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1253–1260, 2010.

[18] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015.

[19] J. Masci, U. Meier, D. Ciresan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Int'l Conf on Artificial Neural Networks*, 2011.

[20] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A.Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 689–696, 2011.

[21] A. Owens, J. Xiao, A. Torralba, and W. Freeman. Shape anchors for data-driven multi-view reconstruction. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 33–40, 2013.

[22] A. Saxena, S.H. Chung, and A.Y. Ng. Learning depth from single monocular images. In *Advances in Neural Information Processing Systems*, 2005.

[23] A. Saxena, M. Sun, and A.Y. Ng. Make3d: Learning 3d scene structure from a single still image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5): 824–840, 2009.

[24] N. Srivastava and R.R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in neural information processing systems*, pages 2222–2230, 2012.

[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[26] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012.

[27] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 702 –709, june 2012. doi: 10.1109/CVPR.2012.6247739.

[28] R. Zhang, S.A. Candra, K. Vetter, and A. Zakhor. Sensor fusion for semantic segmentation of urban scenes. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1850–1857, May 2015. doi: 10.1109/ICRA.2015.7139439.