Deep learning based restoration of lost sections in Micro-CT core plugs

S S Arsenyev-Obraztsov and G O Plusch
Department of Applied Mathematics and Computer Simulation, National University of Oil and Gas “Gubkin University”, 65 Leninsky Prospekt, Moscow, Russia
E-mail: arseniev@gubkin.ru, gpoi@mail.ru

Abstract. Lack of petrophysical information is critical for reservoirs development composed of poorly consolidated rocks or for zones bearing wells with core damaged by improper coring operations. The restoration complexity of the digital-core lost sections is associated with the need to consider an enormous amount of data from the existing core image and the necessity to include lithological expert knowledge. That makes deep learning methods a natural choice for solving such problems. We proposed, examined, and compared several deep learning methods convenient for analyzing micro-computed tomography digital core data. It was done under the most simplistic problem statement when the destroyed part (a set of slices) is completely lost. Here, we present the results of comparison interpolation/extrapolation procedures under proposed quality metrics. We discover that the variational autoencoder method can be trained to extract some petrophysical parameters from the digital core plug in an unsupervised manner.

1. Introduction
Petrophysical parameters are crucial for the characterization of oil and gas reservoirs. The accuracy of the oil field production predicting indicators significantly depends on the amount, reliability, and representativeness of petrophysical data obtained from various sources.

The problem of insufficient petrophysical information arises during the exploration and development of reservoirs composed of poorly consolidated rocks. This situation can also crop out because of incorrect coring operations. Therefore, loose reservoir rocks and intervals with low core recovery are complicated targets for petrophysical examination.

Restoration of the lost or destroyed core sections, depending on the position of the disappeared volume, is the problem of extrapolation or interpolation of the core digital image obtained using micro-computed tomography (Micro-CT). The computational complexity of this task associates with the need to consider an enormous amount of data from the existing parts of the digital core adjusting to the restoring volume. In addition, we need to involve expert knowledge about the potentially admissible core structures, which is problematic. This consideration makes deep learning (DL) methods a natural choice for solving such tasks because of their success in solving various computer vision problems.

1.1. Deep learning and data size
The evident problem in the application of DL methods to x-ray Micro-CT data is its amount. For example, the pack of slices generated from the core plug sinogram usually consists of 1000 images.
1000x1000 pixels each. In the single-precision floating-point format, popular in DL applications, the size of a digital slice is about 4 MB, and the core plug is 4 GB.

Efficient processing of data of this size usually requires deeper networks, which in turn will use even more memory to store weights.

For the DL applications, GPGPU is the most popular hardware, but the limitation on the size of its internal memory becomes the main bottleneck. The solution is to use parallel learning strategies on a set of GPGPUs installed in one node. Unfortunately, in this case, arises the problem of synchronization, which can dramatically slow calculations. A tempting alternative is to use the DL-based Super-Resolution (SR) approach. It was successfully applied to several problems [1]. Using DL-based SR, it is possible to diminish the size of the input data, perform the necessary calculations, and flash the result to SR, returning to the original size without a noticeable accuracy lost due to data compression-decompression. To apply this approach, we need to define the metric of the SR, especially for Micro-CT.

Here we have considered only approaches that allow usage of DL at an original image resolution without involving the SR methods or usage parallelly stackable GPGPUs. The proposed methods were trained in a lower image resolution to diminish computational time.

The source code is available [https://github.com/amcircle/dl-slice-prediction](https://github.com/amcircle/dl-slice-prediction).

1.2. Lost digital core plug sections restoration

Total 3D processing of the digital core plug is the only correct approach for reconstructing the internal porous rock void space. It allows to highlight structural features and generate total vertical and horizontal spatial connectivity between plug’s pores. Also, it excludes the orientation error that occurs during independent 2D processing slices belonging to the digital core pack [2]. For the diminishing calculations and simplicity of the results visualization, we will use the classical method, as processing 2D plug slices will significantly reduce the training time and training complexity. Also, all calculations can be executed by one GPGPU with at least 16GB of local memory.

In Figure 1, we present three principal cases of destroyed or lost core sections. Further, we consider the simplest form of the digital segmented core restoration problem when the destroyed part is completely lost. Cases (a) and (b) are slice image prediction problem (interpolation or extrapolation). Obviously, for case (c), this approach is not proper. For this case, we need to create another strategy that will correctly extract and use information from the preserved core pieces, including a method to restore their position and orientation.

![Figure 1](image-url)  
**Figure 1.** Lost slices presented by red-colored volume. Problems: (a) interpolation, (b) extrapolation, (c) core plug restoration.

We use the binary segmented digital core, as it is much simpler than other image forms. Main essential tasks of petrophysical properties estimation can be conducted on this segmentation [3].
Additionally, grayscale images require more data to train the model resistant to various kinds of noise in the digital image.

At this stage, it is unclear what is the best metric to evaluate the predicted slice. It is possible to use simple pixel-wise metrics like mean squared error (MSE) or mean absolute error (MAE), learnable perceptual metrics or a learned discriminator from adversarial training [4].

We select MSE and mean intersection over union (IoU), which is applicable due to the binary segmentation of image. IoU proved to be helpful in the evaluation of the quality of objects localization and segmentation.

1.3. Dataset and technical aspects

Proposed restoration methods were tested on a synthetic example — reconstruction of deliberately removed slices from a pack of digital core plug.

There are only a few open datasets of digital core samples obtained from Micro-CT. One of the most comprehensive resources that incorporate data about the internal structure of various porous media is Digitalrocks [5]. For training and testing DL methods, we chose dataset [6] consisting of 11 digital core samples. Examples of segmented sections of each core are presented in Figure 2.

![Figure 2](image)

**Figure 2.** Segmented core plug sections: (a) Bandera Brown, (b) Bandera Gray, (c) Buff Berea, (d) Bentheimer, (e) Berea, (f) Berea Sister Gray, (g) Berea Upper Gray, (h) CastleGate, (i) Kirby, (j) Leopard, (k) Parker.

The dataset was split into training and validation samples 80% and 20% of its size. We left Buff Berea slices for testing the overall performance of the proposed methods and the capability of their generalization.

To implement and test proposed DL methods we used Python 3 with TensorFlow 2 library. Neural networks (NN) were built with use of tensorflow.keras layers implementation and the provided architectures presented in Keras notation. The calculations were conducted on Nvidia GPUs P100 and V100 provided by Google Colab.

2. DL slice image prediction

In this work, we consider three different approaches in their basic implementations as we want to know what direction of research is more natural and appropriate for the stated problem.

Since the fact that NN produces continuous outputs, we apply binarization (2.1) on min-max scaled slices as a post-processing technique.
I processed = f \left( \frac{I - I_{\text{min}}}{I_{\text{max}} - I_{\text{min}}} \right), \quad \text{where } f(I) = \begin{cases} 0, & \text{if } I \leq 0.5 \\ 1, & \text{otherwise} \end{cases} \quad (2.1)

2.1. LSTM - Long short-term memory

LSTM \cite{7} (Long Short-Term Memory) is widely used type of recurrent neural networks (RNNs). Further, we evaluate the convolutional form of LSTM \cite{8}. Inspired by \cite{9} we considered the problem of slice image forecast as next frame prediction where the main goal is to predict frames of dynamic visual scenes using information from the past.

We tested different LSTM architectures but failed in training any adequate model. The results were either unsuitable and not applicable for further examination, probably because of underfitting, or the model reproduced only the last slice in the sequence as a prediction (apparently problem of overfitting).

LSTM nets are hard to train due to low parallelization capabilities and large memory consumption. The next frame prediction is usually used to predict deterministic motion on the scene, meaning that such an approach is not appropriate in this case due to the stochastic nature of the core plug slice prediction.

2.2. VAE

Variational Autoencoder (VAE) is a generative model based on variational approximations \cite{10}. The idea of application generative DL models for studying core plug porous media is not new \cite{11}. But we did not find any information concerning the application of VAE for this purpose. VAE describes the latent space in a probabilistic distribution manner rather than producing points in the latent space. VAE learns a probability distribution, which makes this approach attractive considering the quasi-stochastic nature of slices images sequence. The key idea is to make the autoencoder latent space similar to some prior distribution \( p(z) \) and then generate new data using it. The classical VAE represents an autoencoder model, where the encoder maps data sample \( x \) in its latent space representation \( z \) by producing parameters of learned approximate posterior distribution \( q(z|x) \) and sampling from it. The decoder then maps \( z \) to parameters of data likelihood distribution \( p(x|z) \)

\[ z \sim Enc(x) = q(z|x), \quad \tilde{x} \sim Dec(z) = p(x|z). \quad (2.2) \]

In VAE, we regularize the latent space with Kullback-Leibler (KL) divergence, making it similar to a known prior \( p(z) \) which is usually multivariate Gaussian \( \mathcal{N}(0,I) \). Negative log-likelihood represents the reconstruction loss. Then VAE loss (2.3) is the sum of reconstruction loss and the prior regularization.

\[ L_{VAE} = -E_{q(z|x)}[\log p(x|z)] + D_{KL}(q(z|x), p(z)) = L_{\text{rec}} + L_{KL} \quad (2.3) \]

where \( D_{KL} \) is the Kullback-Leibler divergence.

Generating new data samples with VAE comprises picking a random point in latent space, which corresponds to sampling \( \xi \) from some prior distribution and then passing it to the decoder (generator) to produce a sample in data space.

A significant property of VAE is the ability for smooth interpolation of data samples in latent space. It happens because the KL term forces each dimension of the posterior distribution to be concentrated around the origin with a nonzero variance, resulting in overlapping between the distribution of distinct data points. At the same time, the reconstruction term encourages the decoder to produce the intended output. That means that close points in the latent space should generate neighboring data samples in the sense of the chosen reconstruction loss function, which
usually can not be achieved when using a plain autoencoder model. This property makes VAE a manifold learning algorithm.

VAE is straightforward for implementation in different versions of model architectures. We used 2D convolution layers and dense layers to produce the parameters of the posterior distribution. We build an extensive network for faster convergence to test this approach, but the architecture should be carefully designed as large networks are prone to overfitting. The total architecture is shown in Table 1.

| Layer name | Description | Connected to |
|------------|-------------|--------------|
| conv_1     | Conv2D, 32 3x3 kernels, ReLU activation, strides 2 | input |
| conv_2     | Conv2D, 64 3x3 kernels, ReLU activation, strides 2 | conv_1 |
| conv_3     | Conv2D, 128 3x3 kernels, ReLU activation, strides 2 | conv_2 |
| conv_4     | Conv2D, 256 3x3 kernels, ReLU activation, strides 2 | conv_3 |
| flatten    | Flatten 3D array to vector | conv_4 |
| z_mean     | Dense 128, linear activation (output) | flatten |
| z_log_var  | Dense 128, linear activation (output) | flatten |

| Decoder   | Description | Connected to |
|-----------|-------------|--------------|
| dense     | Dense 65536, ReLU activation | input |
| reshape   | Reshape vector to 3D array | dense |
| conv_t_1  | Conv2DTranspose, 256 3x3 kernels, ReLU activation, strides 2 | reshape |
| conv_t_2  | Conv2DTranspose, 128 3x3 kernels, ReLU activation, strides 2 | conv_t_1 |
| conv_t_3  | Conv2DTranspose, 64 3x3 kernels, ReLU activation, strides 2 | conv_t_2 |
| conv_t_4  | Conv2DTranspose, 32 3x3 kernels, ReLU activation, strides 2 | conv_t_3 |
| conv       | Conv2D 1 3x3 kernels, sigmoid activation, strides 1 | conv_t_4 |
| resize     | Bilinear image resize to input width and height (output) | conv |

We trained VAE on 250x250px size slice images with ADAM optimizer, learning rate $10^{-4}$ and batch size 4 for 100 epochs. The example of VAE reconstruction for a sample from the training split (Berea core plug) is shown in Figure 3.

![Figure 3](image.png)

**Figure 3.** Left-to-right: ground-truth slice, VAE reconstruction, binarized VAE reconstruction.

Sometimes the images generated by VAE may be blurry. Probable reasons are model underfitting or incorrect assumptions about the $p(x|z)$ distribution, which leads to incorrect reconstruction loss. When some data samples have multi-modal distribution, VAE cannot
generate images with sharp edges and fine details [12]. To mitigate or overcome this problem, one may use a learnable reconstruction similarity function as proposed in VAEGAN [13] or a special multistage architecture with a refinement network to improve the coarse output from VAE [12].

Latent space analysis may provide some insights into how VAE encodes the multidimensional input information into a vector. Also, it answers the question — whether the independent components of the vector have some interpretable meaning, if it is disentangled or not. Usually, digital core plugs are needed for the evaluation of petrophysical parameters. We selected the porosity $P$ per slice (2.4) as the most straightforward in both guises as a physical and graphical parameter for calculation (Figure 4). Then we mapped each slice to its mean encoding ($z = \mu$, where $\mu$ is the mean of learned Gaussian posterior distribution).

$$P = 1 - \frac{\sum_{n,m} I_{nm}}{nm}$$

(2.4)

To find latent variables similar to porosity, we used correlation distance $c(u, v)$ (2.5) between min-max scaled values of porosity and each latent variable independently.

$$c(u, v) = 1 - \frac{(u - \overline{u}) \cdot (v - \overline{v})}{||u - \overline{u}||_2 ||v - \overline{v}||_2}$$

(2.5)

where $\overline{u}$ and $\overline{v}$ is the mean of elements of vectors $u$ and $v$, $x \cdot y$ is the dot product of $x$ and $y$.

We found that there is more than one latent variable similar to the porosity (Figure 4). VAE highlighted porosity in slices latent space. However, the representation is entangled between different latent variables.

Figure 4. (a) porosity plot, (b)-(d) latent variable (orange) vs porosity (blue) plots.

Checking VAE on the test core plug showed it didn’t generalize well and possessed some overfitting. Nevertheless, we didn’t consider only online slice generation, and we fine-tuned VAE for ten epochs on test core plug slices, except 20, to check if one can use VAE to interpolate
these missing slice images. After the fine-tuning, we encoded all slices of the test core plug with their mean and plotted the 128-dimensional latent space points with PCA and t-SNE (Figure 5).

Figure 5. Left-to-right: PCA and t-SNE test core plug latent space visualization. Each slice point in the latent space is colored in greyscale according to their stack order from white to black. The unseen slices are colored with red.

We can see that core plug latent space points form a direction prediction curve ordered according to slices sequence. It is significant that the unseen slices in the process of training also belong to this curve. Now it is possible to convert the interpolation/extrapolation problem from the image (slice) domain to one-dimensional vector function domain \( g(s) = l \), where \( s \) is a slice index, \( l \) is a latent vector and \( g(s) \) is a function we need to interpolate/extrapolate using produced slices encodings. We used Akima spline \([14]\) to interpolate the encodings for missing slices. The prediction results for the first six slices are shown in Figure 6, the resulting mean \( IoU \) is 0.71857774, and \( MSE \) is 0.32730734.

Figure 6. Example of VAE interpolation application. The first row is ground-truth, the second is reconstruction, the third is prediction. (a) slices, (b) slice patches.

Providing an essentially larger dataset of core plugs makes it possible to use LSTM for encodings interpolation and extrapolation.

It seems that the proposed technique with well generalized VAE can be used for interpolation/extrapolation of a large number of slices. However, it is still unclear whether the predicted slices will have numerically estimated petrophysical parameters such as permeability close to the real ones. Possible solutions are controlling the physics directly by conditional VAE (CVAE) or improving the disentangling ability of the encoder network. This can be achieved, for example, with a weighted KL term in VAE loss function \( \mathcal{L}_{VAE} = \mathcal{L}_{rec} + \beta \mathcal{L}_{KL} \) \([15]\) where \( \beta > 1 \), thought this modification degrades the curve formation of core plug latent representation.
There also exist more sophisticated methods for solving next frame prediction by VAE application [16].

2.3. Self-supervised DL optical flow net
Optical flow estimation is an ill-posed inverse problem solution of evaluating an object’s motion patterns between images of a scene for different time steps. Alternatively, it can be presented as a calculation of the motion of image intensities or pixels [17]. DL methods for optical flow estimation gained rapid development recently and considered being the state-of-the-art methods tested on many benchmarks, for example KITTI-2015 [18]. Optical flow is applicable in many video processing problems; automatic navigation tasks like visual odometry, object detection, tracking, and segmentation [19]. It is also one of the principal research areas in video frames interpolation or next frame prediction.

We describe dense optical flow $flow_{t+k}(x, y) = [\delta x, \delta y]$ between two slices $I^t$ and $I^{t+k}$ as a per-pixel point displacements $\delta x$ and $\delta y$. This pixel interconnection $I^{t+k}(x, y) = I^t(x+\delta x, y+\delta y)$ makes it possible to interpolate slices in-between by interpolating intermediate displacements in optical flow and warping corresponding slices

$$I^{t+r} = w\left(I^t, g(r; 0, flow_{t+k})\right), \quad (2.6)$$

where $t < r < t + k$, $I^t$ is the reference slice, $I^{t+k}$ is the target slice, $I^{t+r}$ is an intermediate slice between the reference and target slices, $w(I, flow)$ is some warping procedure and $g(r; a, b)$ is interpolated value at $r$ between points $a$ and $b$.

The lack of ground truth requires training the network in an unsupervised or self-supervised manner. We propose to use the reconstruction error between the warped frame $\tilde{I}^{t+k}$ and the target frame $I^{t+k}$ as the loss function (2.7).

$$\mathcal{L}_{of} = -E_{u+k}[\log \tilde{I}^{t+k}]. \quad (2.7)$$

To train the network by means of stochastic gradient descent methods $w$ must be differentiable. We will use realization of differentiable sampling [20] from `tensorflow.graphics.image.transformer.sample` module with a bilinear kernel. For simplicity, we set $g(x; a, b)$ as a linear interpolation. Here we do not consider an occlusion problem, as we suppose that the number of pixels in the reference slice is always enough to warp it into the target slice and vice versa. In Table 2 we present total architecture.

### Table 2. Optical flow net architecture.

| Layer name | Description | Connected to |
|------------|-------------|--------------|
| concatenate | Concatenate two input slices channels wise | input_1, input_2 |
| conv_1     | Conv2D, 32 7x7 kernels, ReLU activation, strides 2 | concatenate, conv_1 |
| conv_2     | Conv2D, 64 5x5 kernels, ReLU activation, strides 1 | conv_1 |
| conv_3     | Conv2D, 128 3x3 kernels, ReLU activation, strides 1 | conv_2 |
| conv_4     | Conv2D, 256 3x3 kernels, ReLU activation, strides 1 | conv_3 |
| conv_5     | Conv2DTranspose, 128 3x3 kernels, ReLU activation, strides 2 | conv_4 |
| conv_6     | Conv2D, 64 3x3 kernels, ReLU activation, strides 1 | conv_5 |
| conv_7     | Conv2D, 32 3x3 kernels, ReLU activation, strides 1 | conv_6 |
| conv_8     | Conv2D, 2 3x3 kernels, TanH activation, strides 1 (output) | conv_7 |
We trained optical flow DL net on 250x250px size slice images with ADAM optimizer, learning rate $10^{-4}$ and batch size 4 for 100 epochs. We trained this network on pairs $(I^t, I^{t+k})$, where $k = 21$, so we can compare interpolation results with VAE on the same stack of slices. The example of the warped image and the estimated optical flow visualized with [21] for a pair of samples from the training set (Berea core plug) are shown in Figure 7.

Figure 7. Left-to-right: reference slice $I^t$, target slice $I^{t+k}$, warped slice $\tilde{I}^{t+k}$, optical flow visualization.

Results for test core plug slices interpolation with optical flow presented in Figure 8, the mean IoU is 0.7173934 and MSE is 0.32103884.

Figure 8. Example of optical flow interpolation application. The first row is ground-truth, the second is prediction. (a) slices, (b) slice patches.

We can conclude that linear interpolation is an incorrect choice. Pixel movements look unnatural, meaning that optical flow training needs additional regularization, possibly encourage small displacements. Another problem with the traditional optical flow approach is that unseen structures or formations can not pop up during direct interpolation.

3. Results
As the quality measurement baseline for presented methods, we use the naïve constant first frame interpolation — assignment of the last slice to all predicted $I^{t+k}_{\text{predicted}} := I^{t+k}_{\text{ground-truth}}$. Figure 9 present results of interpolation for the most distant slice (number 10 from 20 slices long interpolation task). Table 3 contains quality metrics for the same slice image.

Figure 9. Left-to-right: ground-truth slice image, VAE prediction, optical flow prediction.
Table 3. Methods comparison.

| Method     | MSE (lower is better) | IoU (higher is better) | Porosity absolute error (lower is better) |
|------------|-----------------------|------------------------|------------------------------------------|
| VAE        | 0.1448                | 0.6546                 | 0.0013                                   |
| Optical flow | 0.1486                | 0.6375                 | 0.0209                                   |
| Naïve      | 0.1547                | 0.6369                 | 0.0024                                   |

Results of VAE and optical flow DL methods are slightly different in MSE. VAE produced two times better results in terms of porosity error in comparison with the naïve approach. We conclude that MSE is an inappropriate metric for problems of this type. IoU quality metric proved to be better than MSE. However, appropriate metrics need to be developed based on the graphic properties controlling results of future physical processes simulations on a digital core plug.

According to Table 3, VAE has the best score in all metrics. Because of the heuristic nature of deep learning methods, further research needs to be conducted with other DL approaches and architectures. Obtained results show that DL and especially VAE are efficient for lost or destroyed core body restoration.

4. Conclusion
The problem of lost Micro-CT scanned data restoration is poorly covered in the existing informational sources. We consider that restoring lost cross-sections in the data domain is a more promising area for research than predicting petrophysical parameters of the lost core plug volume. Restoration of void space and porous media matrix allows researchers to apply a wide range of numerical methods for assessing different petrophysical parameters.

This article proposes DL-based solutions to this problem. It shows that the developed VAE method is the best among those considered in the pixel-wise error of the predicted slices and their petrophysical properties. We need to emphasize the inappropriateness of pixel-wise quality metrics to this specific task, which requires the development of physics-aware metrics and loss functions for DL.

We hope further research improves the quality of lost data restoration and the actual depth of interpolation/extrapolation volume.

References
[1] Wang Z, Chen J and Hoi S C 2020 IEEE transactions on pattern analysis and machine intelligence
[2] Arsenyev-Obraztsov S S, Volkov E A and Plusch G O 2019 IOP Conference Series: Materials Science and Engineering 700 012053 URL https://doi.org/10.1088/1757-899x/700/1/012053
[3] Pak T, Archilha N, Mantovani I and Butler I 2019 Scientific data 6 ISSN 2052-4463
[4] Yang Q, Yan P, Zhang Y, Yu H, Shi Y, Mou X, Kalra M K, Zhang Y, Sun L and Wang G 2018 IEEE transactions on medical imaging 37 1348–1357
[5] Prodanovic M, Esteva M, Hanlon M, Nanda G and Agarwal P 2015 Digital Rocks Portal
[6] Neumann R, Andreeta M and Lucas-Oliveira E 2020 11 sandstones: raw, filtered and segmented data http://www.digitalrocksportal.org/projects/317
[7] Gers F A and Schmidhuber J 2000 Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium vol 3 (IEEE) pp 189–194

[8] Xingjian S, Chen Z, Wang H, Yeung D Y, Wong W K and Woo W c 2015 Advances in neural information processing systems pp 802–810

[9] Srivastava N, Mansimov E and Salakhudinov R 2015 International conference on machine learning (PMLR) pp 843–852

[10] Kingma D P and Welling M 2014 Auto-encoding variational bayes (Preprint 1312.6114)

[11] Mosser L, Dubrule O and Blunt M J 2017 arXiv preprint arXiv:1704.03225

[12] Cai L, Gao H and Ji S 2017 Multi-stage variational auto-encoders for coarse-to-fine image generation (Preprint 1705.07202)

[13] Larsen A B L, Sønderby S K, Larochelle H and Winther O 2016 Autoencoding beyond pixels using a learned similarity metric (Preprint 1512.09300)

[14] Akima H 1970 Journal of the ACM (JACM) 17 589–602

[15] Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S and Lerchner A 2017 ICLR

[16] Bhagat S, Uppal S, Yin Z and Lim N 2020 European Conference on Computer Vision (Springer) pp 102–117

[17] Bovik A C 2009 The essential guide to image processing (Academic Press)

[18] Menze M, Heipke C and Geiger A 2018 ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)

[19] Fortun D, Bouthemy P and Kervrann C 2015 Computer Vision and Image Understanding 134 1–21

[20] Jaderberg M, Simonyan K, Zisserman A and Kavukcuoglu K 2016 Spatial transformer networks (Preprint 1506.02025)

[21] Baker S, Scharstein D, Lewis J, Roth S, Black M J and Szeliski R 2011 International journal of computer vision 92 1–31