Deep Learning
Assignment 2

Luuk van Keeken s3512290
Ayça Avcı s4505972
Sytse Oegema s3173267
Koen van Schuijlenburg s4181085

University of Groningen, Department of Artificial Intelligence.
April 2022

Abstract

Computer Vision and Natural Language Processing are two major fields of research in which more and more deep learning techniques are being applied. Although deep learning models can be very powerful, training them can be costly both in time and energy. This project is aimed at exploring the possible benefits of using a pre-trained model instead of training a model with random initial weights. Comparisons between these different types of initializations were made for an HRNet and a DistilBERT model, within an image segmentation task and a question answering task respectively. It was found that a pre-trained model can reach higher performance levels in fewer epochs of training, which saves time and decreases energy consumption.

1 Introduction

Now that increasingly more data in the form of text and images is freely available, there is an enormous interest in techniques for analyzing this data and utilizing it for all kinds of purposes. For this reason, more and more research has been performed in the fields of Computer Vision (CV) and Natural Language Processing (NLP). As Deep Learning (DL) has for some time now been established as a powerful machine learning techniques, it is also being applied to a wide range of subjects such as CV and NLP, as well as time series analysis [12] and reinforcement learning [1].

Even though in general a lot of data is available today, it can be difficult to gather enough data for specific learning tasks. The current project is aimed at exploring the possible benefits of using a pre-trained model instead of training a model from scratch. For one CV learning task, and one NLP learning task, a comparison is made in terms of the training process and performance of a pre-trained model and a structurally equivalent model with random initial weights.

One task in the field of CV is image segmentation, in which an image is partitioned into the several meaningful segments or objects present. This task is relevant for a large number of applications, ranging from medical imaging [8] to autonomous driving [7]. In the field of autonomous driving, an important aspect is to gain an understanding of the traffic scene around the vehicle. Information about the vehicle's surroundings can be gathered through various types of sensors, such as visual and thermal cameras, and radars. The current project focuses on the semantic segmentation of RGB images of traffic scenes, in which each individual pixel is classified as belonging to, for example, a car, the road, a traffic sign, etc. The Cityscapes dataset [3], which is a popular benchmarking dataset focused on urban traffic scenes, was used to train and validate the two models. A high-resolution network (HRNet) was selected for the comparison, as several HRNets have reached competitive results on the Cityscapes dataset.

The other task implemented is Question Answering (QA). It is a field in NLP that typically includes Extractive QA, Abstractive QA, Multiple-choice QA, Multiple-choice QA with context and Yes-No QA (e.g. producing an answer without knowing the context, extracting an answer from the given context, choosing an answer from multiple-choice, giving a yes-no answer to the question). There are several popular benchmark datasets such as HotPotQA [26], WikiQA [24], bAbI [23], TriviaQA [10], and

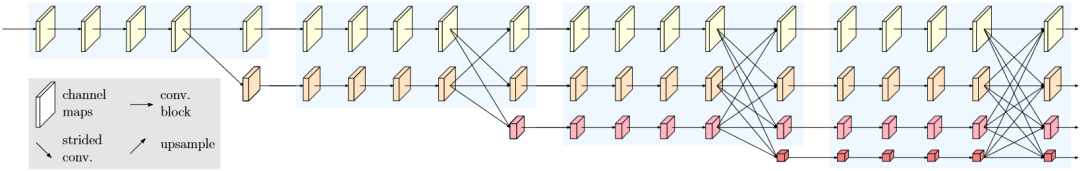

Figure 1: An example of an HRNet consisting of four stages (Microsoft 2020).

SQuAD [19] to evaluate QA systems. Some of the popular and top performing models for QA are BERT [5], T5 [18], XLNet [25] and BigBird [28].

In this report, we presented the results of both a pre-trained and a trained-from-scratch DistilBERT model for a Extractive QA task, which consists of extracting an answer to a given question from the provided context. DistilBERT is a distilled version of BERT which contains 40% less parameters, meaning that it is computationally less expensive than the BERT model, and 60% faster while performing over 95% of BERT's performance according to the GLUE language understanding benchmark. One of the most popular benchmark datasets, the Stanford Question Answering Dataset (SQuAD), is used for training DistilBERT model. To evaluate the model performance, the EM and F1 score metrics are used.

2 Related work

Encoder-decoder based models are a popular type of deep learning networks used in segmentation tasks [14]. These types of models, of which DeConvNet [16] and SegNet [2] are examples, generally have an encoder part consisting of convolutional layers, and a decoder part consisting of deconvolution and unpooling layers. The network then outputs an image as the same size as the input image, containing a pixel-wise classification. A disadvantage of these networks is that the encoding process reduces the resolution of the input image, while reaching the best performance in visual tasks such as segmentation requires detailed and spatially precise representations [13].

An HRNet is a type of architecture in which a different approach is taken. Instead of having to recover a high-resolution representation from a low-resolution representation, a high-resolution representation is maintained throughout the network (see Figure 1). The network consists of several stages, and at each stage a lower-resolution convolution stream is added in parallel. At each stage, through strided convolutions and upsampling multi-resolution fusions are performed, using the lower-resolution representations to enrich the higher-resolution representations and vice versa. Because of this fusion scheme, and because the high-resolution representation is maintained throughout the network, the learned representations are more detailed and semantically rich. Because of these qualities, HRNets reach state-of-the-art performance in several computer vision tasks, such as human pose estimation, object recognition, and semantic segmentation [22].

With a class IoU of 84.5, the best performing HRNet in the Cityscapes' Pixel-Level Semantic Labeling Task benchmark [3] is HRNetV2 in combination with Object-Contextual Representations (OCR) and SegFix. The model has 48 channels, and was pre-trained on the ImageNet dataset [4] and on the Mapillary Vistas dataset [15][27]. Due to the limited amount of available training time, it was decided to fine-tune the HRNet-W18-C-Small-v1 instead, without applying OCR and SegFix. For this network size, there is a model pre-trained on the ImageNet dataset available [22].

NLP is a broad field that tries to fill the gap between computerized artificial understanding and natural languages. NLP includes numerous tasks, such as text classification, information extraction, sentiment analysis, and question answering [20]. Due to this variety in tasks there also exists a variety of NLP frameworks and techniques that can be used. For a lot of these NLP tasks the encoder-decoder based models with an attention mechanism has become very popular[9]. An attention mechanism is a simple method that scores each element in the sequence which enables the underlying network to pay attention to the higher scored elements. Specifically, self-attention can be used in NLP to create a word embedding for the tokens in the text sequence. An example architecture that uses self-attention is the Transformer [21]. The Transformer is used in the two popular NLP frameworks Bidirectional Encoder Representation of Transformer (BERT) and Generative Pre-trained Transformer (GPT) [6][17].

Both BERT and GPT use the Transformer architecture to pre-train a network that has a basic un-

Figure 2: Train image of the Cityscapes dataset in Zurich, Germany [3].

Figure 3: Ground truth training annotation of the Cityscapes dataset in Zurich, Germany [3].

derstanding of natural language. The pre-training phase includes a sequence of self-supervised tasks that are used to build contextual representation of context sequences[21]. After the pre-train phase an additional layer can be added to both of the networks to perform a certain NLP related task. This final layer is fine tuned to that specific NLP task in a supervised learning approach. Both models are used a lot due to their universal application by means of using different final layers.

3 Methods

3.1 Segmentation

3.1.1 Dataset

The Cityscapes dataset [3] consists of traffic scenes in fifty German cities. The dataset is separated in subsets of 2975 training, 500 validating, and 1525 testing images. All images are captured at daytime in good/medium weather conditions, as shown in Figures 2 and 3. For an unbiased validation and testing, the images in the three subsets are all from different cities. The annotations include thirty classes, encoded as colors, which are shown in Table 1, and Table 2. The ignored classes in Table 2 are not annotated in the evaluation subset, because they are considered as unimportant, and therefore background. Optionally, the Cityscapes dataset supplies 19998 training and validation images with coarse annotations, which can be used together with the default dataset.

Obviously, the dataset is not balanced (e.g. a traffic sign is much smaller than a car, or some objects appear more often than others). The class weights to compensate the imbalance is already calculated [22], and shown in Table 1.

3.1.2 Model

To see the effect of using a pre-trained model, two networks with the same architecture are trained: one model which has the initial weights sampled from the normal distribution, and one model which was pre-trained on the ImageNet dataset. It was expected that the pre-trained model outperforms the randomly initialised model because it has seen more training examples.

Because fine-tuning the pre-trained model is computationally expensive, it was not feasible to perform an extensive grid search. The hyperparameters were adapted from the process of training the HRNetV2-W48, which includes an image size of 1024×512, a scale factor of 16, and using multi scale. The initial learning rate was 0.01, and an exponential learning rate schedule was implemented (see Equation 1).

$$\text{lr} = 0.01 \left(1 - \frac{\text{epoch}}{484}\right)^{0.9} \tag{1}$$

The weight decay was set to 0.0005, and the momentum decay factor to 0.9. The model was trained using mini-batch gradient descent. The batch size was set to 16, as this is the highest batch size which fits in the memory of the used GPU (GeForce RTX 3080 Ti).

The best performing model (HRNetV2 + OCR + SegFix) was trained on the Cityscapes training set for 484 epochs. This number of epochs was also used in training the two networks in the current project. Because of time constraints, the models were only evaluated on the validation set after every

Table 1: Included classes Cityscapes dataset

Label	Class weight
road	0.8373
sidewalk	0.918
building	0.866
wall	1.0345
fence	1.0166
pole	0.9969
traffic light	0.9754
traffic sign	1.0489
vegetation	0.8786
terrain	1.0023
sky	0.9539
person	0.9843
rider	1.1116
car	0.9037
truck	1.0865
bus	1.0955
train	1.0865
motorcycle	1.1529
bicycle	1.0507

Table 2: Ignored classes Cityscapes dataset

Label
static
dynamic
ground
parking
rail track
guard rail
bridge
tunnel
polegroup
caravan
trailer

twenty epochs. The two models were compared based on their best validation accuracies. Because the Cityscapes test set is not directly available, a benchmark score could not be obtained.

3.2 Question Answering

3.2.1 Dataset

The SQuAD dataset contains approximately 108K data samples that were generated from 536 of the top 10K Wikipedia articles. It contains columns 'id', 'title', 'context', 'question', and 'answers', which correspond to the id of the text used, title of the text, the text itself, the question regarding the text, and possible answers. The answer to each question is a part of the text provided in the 'context' column. Table 3 represents a portion of the SQuAD dataset. The dataset is divided into 87599 training, 10570 validation, and 9616 test samples, which is approximately 81%, 10% and 9% of the whole dataset respectively.

Table 3: 4 data samples in SQuAD dataset.

id	title	context	question	answer
5726...	Korean War	Against the rested...	Where did the...?	Yokohama, Japan
56df...	Oklahoma City	On December 2009, ...	When was MAPS 3...?	December 2009
5730...	Antenna (radio)	Polarization is the sum...	What is the imagenary...?	radio wave
571b...	Asphalt	The word asphalt is...	From what language...?	Greek

3.2.2 Model

To see the effect of using a pre-trained model, two networks with the same architecture are trained for the QA task: a DistilBERT model for which the initial weights were samples from a truncated normal distribution with standard deviation 0.02, and the pre-trained DistilBERT model, which is trained on a combination of English Wikipedia and Toronto Book Corpus [29]. The pre-trained DistilBERT model is fine-tuned on the SQuAD dataset. Since fine-tuning the pre-trained model is computationally expensive, we have not fine-tuned the hyper-parameters, and went with the default values as suggested in its configuration file. The batch size was set to 16, and the learning rate was set to 2e-5 initially inside

Table 4: IoU per class

Label	pre-trained	randomly initialized
road	97.6	97.1
sidewalk	82.0	78.3
building	90.8	89.3
wall	44.7	41.1
fence	52.2	46.4
pole	59.2	53.8
traffic light	61.7	53.6
traffic sign	73.5	66.6
vegetation	91.8	91.3
terrain	61.3	58.1
sky	93.8	93.2
person	77.8	72.7
rider	52.1	45.7
car	92.7	91.6
truck	48.9	40.3
bus	71.8	59.9
train	45.7	26.5
motorcycle	49.7	39.1
bicycle	72.6	68.4

the optimizer for both models. Since all Transformer models compute the appropriate loss for their task internally, we did not specify any loss function. The model trained from scratch is trained for 4 and 10 epochs. The pre-trained model is fine-tuned for 2 and 4 epochs. Both models are trained on the Peregrine Cluster of the RUG. The two models were compared based on their training and validation loss. The test scores from both models are compared using the Exact Match (EM) and F1 score evaluation metrics.

4 Results

4.1 Segmentation

For both the pre-trained model and the model with random initial weights, the best validation performance was reached after 484 epochs. The pre-trained model obtained a mean IoU of 69.5%, while the randomly initialised model obtained a mean IoU of 63.8%. The mean IoU scores per class for each network can be found in Table 4. For the pre-trained network, the lowest IoU is achieved for the 'wall' class (44.7%), and the highest IoU is achieved for the 'road' class (97.6%). For the model with random initial weights, the lowest IoU is achieved for the 'train' class (26.5%), and the highest IoU is achieved for the 'road' class as well (97.1%).

The progress of the training and validation losses are visualized in Figure 4, and the validation scores are visualized in Figure 5.

Both networks have trained for 484 epochs, and each model was validated 24 times at intervals of 20 epochs. The pre-trained model took 22 hours to complete, and the randomly initialised network took 24 hours and 20 minutes.

4.2 Question Answering

Both the pre-trained model and the model with random initial weights were trained in batches of 2 epochs at a time since the training time of a single epoch took 1 hour approximately on the GPUs of the RUG's Peregrine Cluster[1]. The model with random initial weights has been trained for a longer period since the training loss was still decreasing, whereas the results of the pre-trained model seemed to remain steady. The training and validation score of both models can be found in Figure 6.

[1] https://www.rug.nl/society-business/centre-for-information-technology/research/services/hpc/facilities/peregrine-hpc-cluster

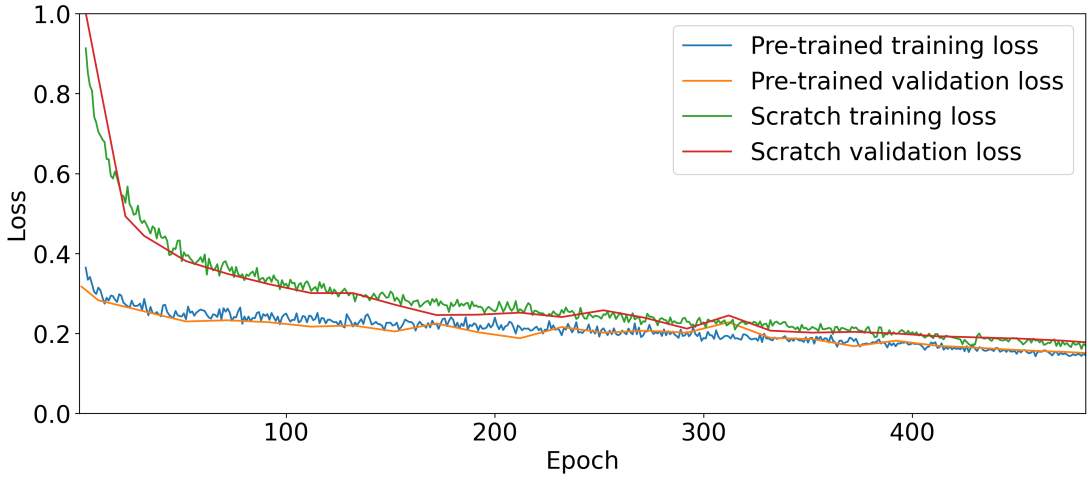

Figure 4: Training and validation loss for both models

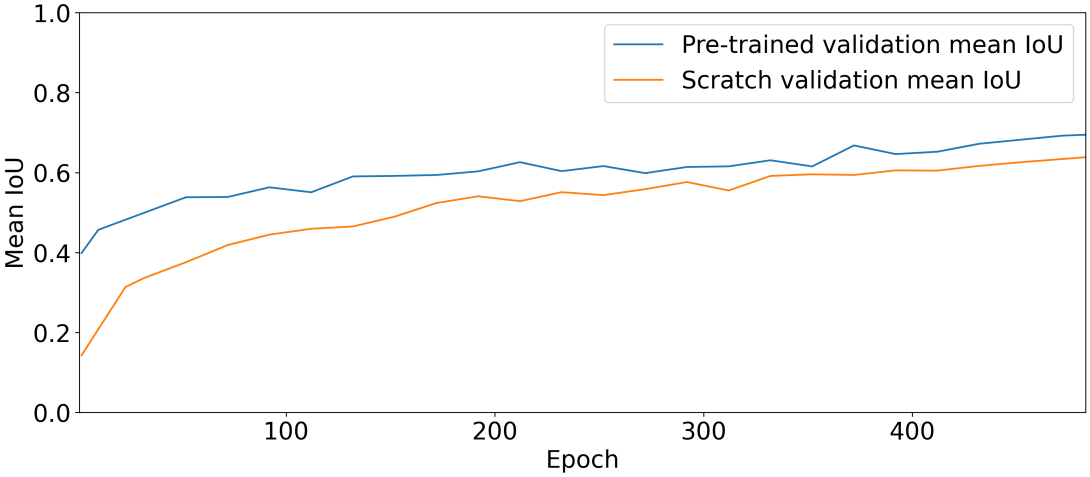

Figure 5: Validation mean IoU for both models

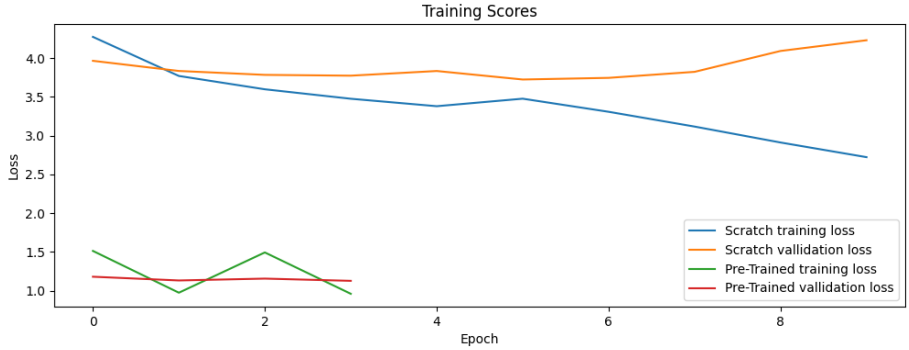

Figure 6: NLP training and validation loss for both models.

Table 5: NLP final test scores.

Model	Training epochs	Exact Match	F1
Trained from scratch	4	13.2%	20.71
	10	12.3%	19.80
Fine-tuning of pre-trained	2	82.3%	88.28
	4	82.5%	88.83

After each batch of training, the weights were saved meaning that for every 2 epochs, a set of weights is available, and they can be used for making predictions. Table 5 shows the prediction scores of the best performing pre-trained and random initial weights model, and both models at their final training epoch (respectively 4 and 10 epochs). Additionally, a sample from the test set is displayed in NLP-sample 4.1. In the sample, the true answer has been marked in **bold text**, and the prediction of each model has been marked with a color.

Context : Lindisfarne are a folk-rock group with a strong Tyneside connection. Their most famous song, "Fog on the Tyne" (1971), was covered by Geordie ex-footballer Paul Gascoigne in 1990. Venom, reckoned by many to be the originators of black metal and extremely influential to the extreme metal scene as a whole, formed in Newcastle in 1979. Folk metal band Skyclad, often regarded as the first folk metal band, also formed in Newcastle after the break-up of Martin Walkyier thrash metal band, Sabbat. Andy Taylor, former lead guitarist of **Duran Duran** was born here in 1961. Brian Johnson was a member of local rock band Geordie before becoming the lead vocalist of AC/DC.
Question : What group is Newcastle native Andy Taylor the former lead guitarist of?
Legend : **True answer** , pre-trained model , scratch model 4-epochs , scratch model 10-epochs ,

NLP-sample 4.1: Sample question from the test dataset with the true and predicted answers.

5 Discussion

5.1 Segmentation

Regarding the training and validation losses, what stands out immediately is that the loss of the model trained from scratch starts out significantly higher than the loss of the pre-trained network. As training continues, the losses of the pre-trained network decrease, but not by much. The losses of the randomly initialized network do approach the losses of the pre-trained network, but this does take some time.

The validation mean IoUs of the two networks follow a similar pattern as they progress along the epochs. The mean IoU of the network trained from scratch starts out lower, but does approach the mean IoU of the pre-trained network. This pattern in the losses and the validation IoUs reveal the downside of training a network initialized with random weights, in the sense that more epochs are required to reach the performance level of a pre-trained network.

For both networks it can be observed that the losses and the mean IoUs have not converged yet. This is not surprising, as the number of epochs was taken as a rough estimate based on the number of epochs used in training HRNet + OCR + SegFix. It would be interesting to see what the result would be of longer training, not only to know the individual performance levels that the models can reach, but also whether or not the model with random initial weights is able to catch up with the pre-trained network at some point. Based on the available data it can only be concluded that the model trained from scratch requires more training epochs to reach the same performance. Additionally to training for more epochs, the model performances could be further increased by also training on the coarse dataset of Cityscapes.

Due to time constraints it was not possible to validate the model after each epoch. Validating after each epoch would have allowed for a more precise view of the developments of the validation losses. However, the plots still show consistent downwards trends in the losses, indicating that it is unlikely that any important information has been missed.

Regarding the IoU scores per class, it was observed that the scores are on a broad range. For all

classes, the randomly initialized network performs worse than the pre-trained network. In general, the classes for which a low IoU is achieved are less frequently represented in the dataset. To counteract this, the class weights as presented in Table 1 could have been used. However, the specific implementation for this in the authors' code required the used of older GPU driver. Because adjusting those drivers is a complicated process, and because the project was not aimed at achieving the best possible performance levels, it was decided to not make use of the class weights. In a project centered around designing the best possible model, including the class weights to counteract the imbalance in the dataset would definitely be required.

5.2 Question Answering

The training and validation loss of the model with random initial weights steadily decrease at the start (see Figure 6). The losses do not decrease drastically anymore after the first epoch which is probably due to the large number of training samples. Especially the validation loss remains very steady up until around the 6th epoch, after which the validation error starts to increase. So the model seems to be overfitting after the 6th epoch, because the training error keeps decreasing, and the validation error gently increases. Therefore, optimal model with random initial weights was obtained at epoch 4.

There is a striking difference between the losses of both the models. The losses of the model with random initial weights are around 4 times larger than the losses of the pre-trained model. Although the pre-trained model was only trained for 4 epochs to save computational power, the validation loss seems to be more stable as well. This performance difference also becomes clear from the test scores of the model predictions on the test dataset. As shown in Table 5, the pre-trained model reaches over 82% of exact matches while the model with random initial weights cannot reach more than 14%. The test sample that is shown in Section 4.2 indicates the difference between the models as well. The pre-trained model predicts the correct answer, but the models with random initial weights provides answers that do not answer the question or that do not even make sense at all.

From the experiment, it is evident that the pre-trained model outperforms the model that is trained with random initial weights since it has been trained on a bigger dataset. In order to achieve better performance on model trained from scratch, more epochs are needed for the training. Since it was computationally expensive and time was limited, we could not explore how the model accuracy might improve. With better computational resources this would have been possible in a given time frame.

6 Conclusion

In both the CV task and the NLP task it became clear that the use of a pre-trained network can provide several benefits. The most important downside of training a model from scratch is that more training epochs are required to reach the performance of a pre-trained network, if reaching that level is even possible. Having to train for fewer epochs not only saves time, but also decreases energy consumption. It should be mentioned that even though the energy consumed during training can be decreased when using a pre-trained network, much more energy is actually consumed when the network is taken into use [11]. This makes sense, as in many cases a trained network is deployed by a large number of uses or in a large number of devices, eventually resulting in a much larger number of computational operations than were performed during training. Thus, regarding energy consumption, much more progress can be made than just using pre-trained networks.

An additional benefit of using a pre-trained network is that less training data is required to reach the same performance levels. Particularly for the segmentation task this is very useful, as it is very labour-intensive to label the training data.

7 Individual contribution

Koen and Luuk worked on the Computer Vision task. Ayça and Sytse worked on the Natural Language Processing task.

References

[1] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Fabian Duffhauss, Claudius Gläser, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *CoRR*, abs/1902.07830, 2019.

[8] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. Deep learning techniques for medical image segmentation: achievements and challenges. *Journal of digital imaging*, 32(4):582–596, 2019.

[9] Dichao Hu. An introductory survey on attention mechanisms in nlp problems. In *Proceedings of SAI Intelligent Systems Conference*, pages 432–448. Springer, 2019.

[10] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017.

[11] Stathis Kamperis. Energy considerations for training deep neural networks, Aug 2019.

[12] Martin Längkvist, Lars Karlsson, and Amy Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24, 2014.

[13] High-resolution network: A universal neural architecture for visual recognition, Jun 2020.

[14] Shervin Minaee, Yuri Y. Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[15] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017.

[16] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[17] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[18] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

[19] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[20] Amirsina Torfi, Rouzbeh A Shirvani, Yaser Keneshloo, Nader Tavaf, and Edward A Fox. Natural language processing advancements by deep learning: A survey. *arXiv preprint arXiv:2003.01200*, 2020.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[22] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *CoRR*, abs/1908.07919, 2019.

[23] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks, 2015.

[24] Yi Yang, Wen-tau Yih, and Christopher Meek. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[25] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.

[26] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.

[27] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *CoRR*, abs/1909.11065, 2019.

[28] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. Big bird: Transformers for longer sequences. 2020.

[29] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27, 2015.