Relevant 2-Step Pretraining Improves Semantic Segmentation on Limited Data

Tau Merand, Richard Klein

Abstract

Pretrained models have been shown to be an effective starting point for training new models. In this work we show that doing a 2-step pretraining process improves semantic segmentation on small COVID-19 x-ray dataset. This 2-step process uses both image recognition and object localisation based x-ray datasets to pretrain on. This offers benefits in allowing segmentation datasets to be small as long as larger but easier to label data is available for pretraininig.

1 Introduction

Semantic segmentation of images is a complex task that has many useful applications (Long, Shelhamer, and Darrell 2015). In recent years deep learning, in particular deep convolutional neural networks have become increasingly effective at this task (Chen et al. 2014). However the cost and difficulty in labelling semantic segmentation datasets is often prohibitive. While large benchmark segmentation datasets, such as CityScapes (Cordts et al. 2016) are available, often novel or niche use cases are impossible for lack of labelled data. Chest x-rays datasets are reasonably common and generally large (Irvin et al. 2019), however these are usually simple image recognition datasets and can provide little clinical benefit or interpretability since abnormalities are not localised for medical professionals to review. Recently a dataset of COVID-19 x-rays with abnormalities localised by bounding boxes was created for a Kaggle competition (Lakhani et al. 2021). However, despite the topicality of COVID-19, the largest segmentation dataset found at the time of writing only contains 100 annotated images. In this paper we demonstrate that using a 2-step pretraining process involving relevant recognition and localisation datasets improves semantic segmentation on a related small dataset.

2 Related work

Pretraining or transfer learning has become increasingly common, particularly in convolutional neural networks. Weights or filters trained on ImageNet (Deng et al. 2009) have become a standard network initialisation strategy. It has been shown that the convolutional filters learnt by training on ImageNet are effective at image processing and feature extraction (Erhan et al. 2010). However the efficacy of transfer learning or pretraining is generally related to the closeness between the domains. Weights trained for a specific task will generally not transfer as beneficially to a completely unrelated task as it would to a more closely related one (Zamir et al. 2018).

It is normal for semantic segmentation networks to be built around a "backbone" network architecture that has been proven to be effective at some other vision task, usually image recognition or classification (Arnab, Miksik, and Torr 2018). For example DeepLabv3 was originally created with ResNet as its backbone (Chen et al. 2017). However while semantic segmentation is often related to image recognition it is not guaranteed that transfer learning will be beneficial to a particular segmentation task (Hong et al. 2016).

3 Methodology

3.1 Data

COVID-19 Chest X-ray Segmentation dataset: CCXS is a dataset that consists of only 100 x-rays of COVID-19 positive adults which that have been semantically segmented by volunteers. As shown in Table 1 CCXS has 13 segmentation categories. All images in CCXS have the normal anatomical segmentations of lungs, cardiomediastinum (basically the heart), airways and central veinous line as well as at least one of the COVID-19 related pathological segmentations: ground glass opacities, pleural effusion or consolidation. Various medical devices visible in the x-rays are also segmented but are not present in all images.

The test set for training on CCXS consists of 10 images selected such that all segmentations, excluding pneumothorax, were present in at least one image. This is possible since each image has multiple segmentations while the single image with pneumothorax remained in the training set. The remaining 90 images were used as for training and validation as discussed in Section 3.3. This test set was held out until the end of this project and only used for the final results seen in Table 2. No experiments or hyperparameter tuning was done using this test set.

SIIM-FISABIO-RSNA COVID-19 dataset: SFRC is a dataset created for a Kaggle competition in 2021, provided by the Society for Imaging Informatics in Medicine (SIIM), the Foundation for the Promotion of Health and Biomedical

Category	
Right Lung	100
Left Lung	100
Cardiomediastinum	100
Airways	100
Ground glass opacities	93
Consolidation	32
Pleural effusion	2
Pneumothorax	1
Endotracheal tube	13
Central veinous line	11
Monitoring probes	26
Nasogastric tube	10
Other tubing	16

Table 1: Summary of segmentations of CCXS

Pathology	Pos.(%)	Uncertain(%)	Neg.(%)
No Finding	8.86	0	91.14
Enlarged Cardiom.	4.81	5.41	89.78
Cardiomegaly	12.26	3.52	84.23
Lung Lesion	3.65	0.57	95.78
Lung Opacity	49.39	2.31	48.3
Edema	26.06	6.17	67.77
Consolidation	6.78	12.78	80.44
Pneumonia	2.44	8.34	89.22
Atelectasis	5.6	15.66	68.71
Pneumothorax	9.23	1.42	89.35
Pleural Effusion	40.34	5.02	54.64
Pleural Other	1.3	0.94	97.76
Fracture	3.87	0.26	95.87
Support Devices	56.4	0.48	43.12

Table 2: Summary of labels in CheXpert

Research of Valencia Region (FISABIO), and the Radiological Society of North America (RSNA). It consists of 7597 images that have been annotated by radiologists with bounding boxes around COVID-19 pneumonia related abnormalities. These bounding boxes are labeled with one of: negative for pneumonia, typical appearance, indeterminate appearance or atypical appearance. Each image can and usually does have multiple bounding boxes however $< 1\%$ have different labels for each bounding box within an image. X-rays with no abnormalities are denoted with a 1x1 pixel bounding box labeled negative for pneumonia.

The SFRS dataset was split 70:15:15 into train, validation and test sets. However care was taken to insure that individual patients with multiple x-rays only appear in a single subset to prevent data leakage between the train/validation/test sets. The 4 labelled classes, while not perfectly balanced, are balanced enough that no additional sampling methods were used.

CheXpert dataset: CheXpert is a dataset consisting of 224,316 chest x-rays created using an automated labeler that extracted observations from medical chest radiographic studies done at Stanford Hospital, between October 2002 and July 2017 (Irvin et al. 2019). The labeler used was trained against x-rays labelled by teams of radiologists and at the time of writing is considered to be the most accurate, large chest x-ray dataset available. Each x-ray in CheXpert was labeled with either positive, negative or uncertain for the presence of 14 common observations as shown in Table. A x-ray can be labelled with more than one pathology, though positive for no finding is mutually exclusive from any of the other labels.

As suggested in (Irvin et al. 2019) images with uncertain labels for atelectasis, edema and pleural effusion were relabelled as positive for those categories while uncertain for all categories were relabelled as negative for those categories. An accurate prediction on an image from the CheXpert dataset is simply a correct multiclass label containing all the positive observations for that image.

The CheXpert dataset was split 70:15:15 into train, validation and test sets. Again care was taken to insure that individual patients with multiple x-rays only appear in a single subset to prevent data leakage. The CheXpert dataset is actually quite imbalanced between classes however since it is large enough that the validation and test sets easily contain all the different classes and as discussed in Section 3.3 it is used for pretraining only no additional sampling methods were used.

3.2 Network Architectures

Classifiers Three image classification or image recognition architectures are used in this paper: VGG16, ResNet50 and ResNeXt50. VGG16 is a comparatively older network architecture (Simonyan and Zisserman 2014) and while it has been, in many cases, supplanted by newer architectures, in particular ResNet and ResNeXt (Xie et al. 2017), it was selected due to its simplicity and is still one of the standard sequential networks in common use. ResNet (He et al. 2016) and ResNeXt are known to be more effective modern networks, constructed to allow increased network depth in comparison to VGG16 (Xie et al. 2017).

While VGG16 is a sequential network both ResNet and ResNeXt are based around the idea of skip connections within the network. ResNet works off single residual blocks with skip connection between blocks whereas ResNeXt works works off of wide, parallel residual blocks with concatenation or aggregation operations between blocks.

All three of these network architectures are suitable for use as backbones for the segmentation networks used in this paper. In addition since they all are convolutional, image dimension differences between the different datasets are only an issue for the fully connected layers used for final classification. This allows for simple transfer of learnt filters between the different experiments as discussed in Section 3.3. The training of these networks as classifiers is discussed in Section 3.3.

Autoencoders: VGG16, ResNet and ResNeXt were also selected due to the simplicity of implementing autoencoders based on these architectures. An autoencoder is simply a network thats output is the same shape as its input. They are

then trained using the difference between input and output, known as the reconstruction error. Classifiers can be easily converted into an autoencoder by removing the fully connected, output layers then mirroring the network. Care must be taken to upsample any layers of the network, such as pooling, that decrease the dimensions of the signal through the network. The first part of the network is known as the encoder, which usually reduces the dimensionality of the input while the second part consisting of the reversed network, with upsampling, is known as the decoder which returns the input to the original dimensions. This approach is useful since any error in the labelling of CheXpert is not relevant to the reconstruction error. As discussed below this encoder/decoder structure is also very similar to the structure of UNet.

Segmentation Networks This paper uses two segmentation network architectures namely UNet (Ronneberger, Fischer, and Brox 2015) and DeepLabv3 (Chen et al. 2017) since both have been shown to be effective and versatile networks for semantic segmentation.

UNet is a two part network: the encoder which consists of stacks of several convolutional layers followed by a pooling layer then a decoder which is the encoder reversed with pooling replaced by up sampling layers. Between corresponding layers in the encoder and decoder are copy and crop skip connections which concatenate the output of each encoder layer and the input of each decoder layer as illustrated in Figure 1. The encoder layers extract features of different spatial resolutions which are then used by the decoder layers via these skip connections to produce segmentation masks.

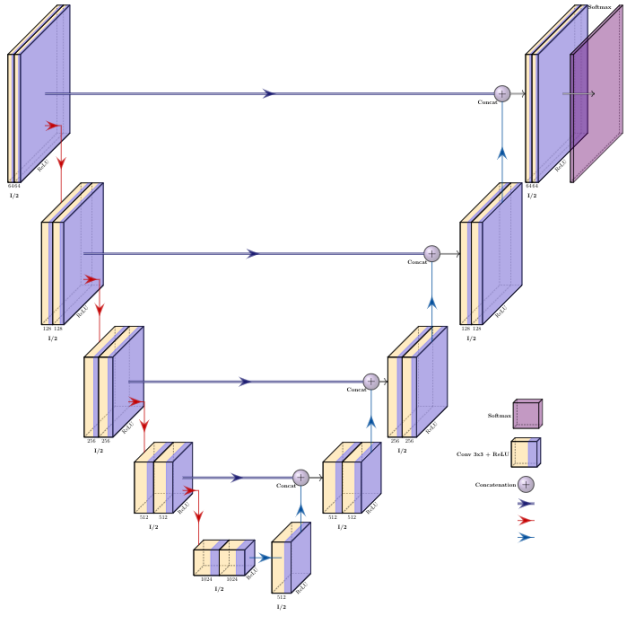

Figure 1: An example of UNet architecture based on a VGG16 backbone. ResNet and ResNeXt are also suitable backbones for UNet.

DeepLabv3 on the other hand only has an encoder followed by an Atrous Spatial Pyramid Pooling (ASPP) module. The encoder of DeepLabv3 consists of convolutional layers with deeper convolutional layers utilising atrous or dilated convolutions. These are convolutions where inputs . These are followed by an ASPP module where atrous convolutions at multiple scales are combined with spacial pyramid pooling to capture a wide range of context for segmentation without greatly increasing the complexity of the network.

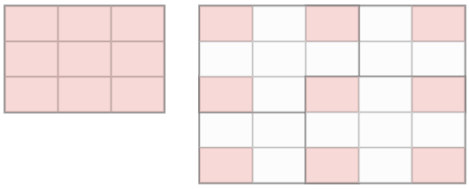

Figure 2: Left: 3x3 Convolution, Right: 3x3 stride=1 Dilated Convolution

Both UNet and DeepLabv3 are capable of using a variety of backbones. In Figure 1 a VGG16 backbone is in use. A backbone in a segmentation network is usually an existing, effective classification network. We used VGG16, ResNet50 and ResNeXt50 backbones.

3.3 Training

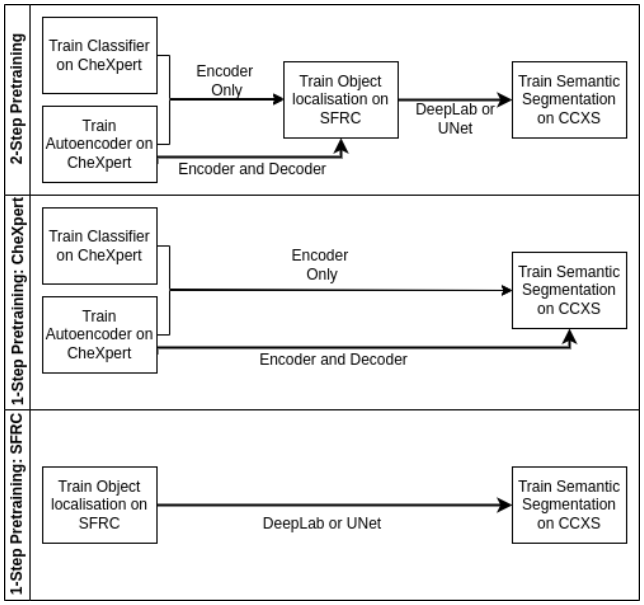

Figure 3: Comparison of 2-step and 1-step pretraining used in this paper.

No pretraining: Since CCXS is so small, semantic segmentation methods struggle to achieve good results without

becoming over trained. No pretraining provides the baseline results for semantic segmentation on limited data. In this no pretraining phase Unet and DeepLabv3 networks were trained using only CCXS. The networks were either intitialised randomly or using ImageNet trained weights for the backbone. UNet and DeepLabV3 were implemented using the Pytorch Segmentation Models library (**?**). Each segmentation architecture was trained with all 3 backbones: VGG16, ResNet50 and ResNeXt50 to early stopping.

Training took place on the 90 image training set from CCXS, using 9-fold cross validation. At the start of each training cycle the 90 image training set is partitioned into nine subsets. A single ten image partition is held out as a validation set, training is performed on the remaining 80 images then the validation error is calculated using the held out set. The next runs select the subsequent partition as the hold out and training is repeated. 9 such runs form an epoch, since the network will have trained on all 90 images and been validated on all 9 partitions. Between each epoch the training set was shuffled so that the partitions vary. To help prevent over training early stopping and basic image augmentations were also used. Image augmentations included mild blur, y-axis reflection and small rotations.

1-Step Pretraining: 1-Step pretraining involves training the network on a single different dataset before training the resulting network on CCXS. Pretraining is thought to be effective by placing the network in a beneficial "place" in the loss landscape before training begins (Erhan et al. 2010). Since we have two datasets in consideration we do 1-step pretraining on both only CheXpert and only SFRC.

Pretraining on CheXpert is possible in two ways: training a classifier or training an autoencoder. For the classifier approach VGG16, ResNet50 and ResNeXt classifiers are trained on CheXpert as an image recognition task to early stopping. Then these trained weights are loaded as the encoder for both UNet and DeepLabv3 and are then trained to do semantic segmentation on CCXS. This replaces the need for weight initialisation for DeepLabv3 and for the encoder portion of UNet, however the decoder portion of UNet is still initialised either randomly or using ImageNet weights. The autoencoder approach trains VGG16, ResNet50 and ResNeXt autoencoders on CheXpert then the weights of the encoder portion of each autoencoder is transferred into DeepLabv3 and either the encoder only or both the encoder and decoder are transferred into UNet.

Despite SFRC being an object localisation via bounding boxes dataset 1-step pretraining on SFRC is done using DeepLabv3 and UNet directly. Each x-ray in SFRC has a mask generated using the bounding boxes. All pixels inside each box is labeled as if it was a semantic segmentation mask. This allows the segmentation networks to be used as if this was a segmentation task. All the different backbones are utilised, and once the networks are trained to early stopping on SFRC the exact same networks are trained on CCXS.

2-Step Pretraining This is overall the most computationally expensive pretraining process as it involves training on CheXpert as either a classifier or an autoencoder then transferring those trained weights to a UNet or DeepLabv3 and training on SFRC an then finally training on the actual segmentation task on CCXS. As shown in Figure 3 2-step pretraining involves training both classifiers and autoencoders on CheXpert then transferring either just the trained encoder or both the encoder and decoder to UNet.

4 Results

As can be seen in figures 4 and 5 both 1-step pretraining and 2-step pretraining provide better segmentation performance on CCXS. 2-step pretraining using an autoencoder for the CheXpert step performs the best. In both DeepLabv3 and UNet ResNeXt50 is the best performing backbone. Overall DeepLabv3 performs better than UNet however of more consideration is the relevant improvement 2-step pretraining offers.

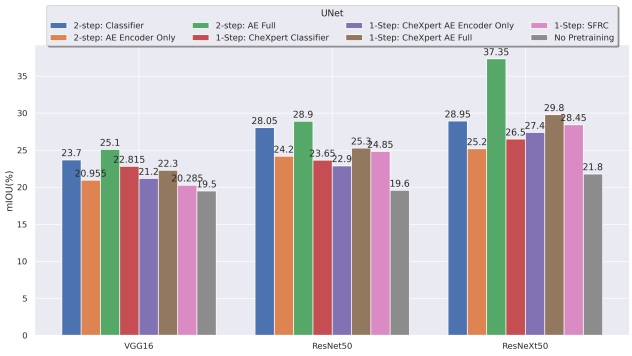

Figure 4: Comparison of mIOU(%) achieved by UNet on the test set of CCXS.Bars are grouped based on the backbone used by UNet. AE refers to the use of an autoencoder for pretraining on CheXpert.

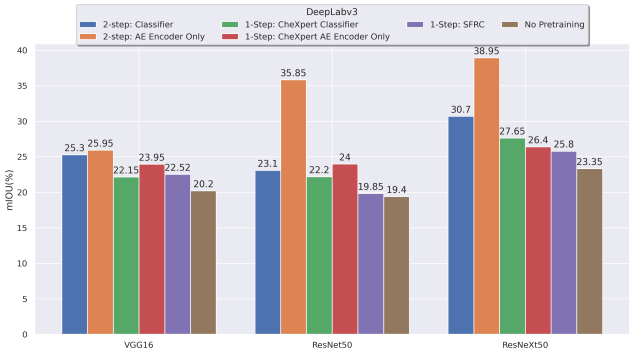

Figure 5: Comparison of mIOU(%) achieved by DeepLabv3 using different backbones on the test set of CCXS. AE refers to the use of an autoencoder for pretraining on CheXpert.

As can be seen in Table 3 2-step pretraining achieves a higher mIOU across all the models. In particular DeepLabv3 achieves 39.6% mIOU when trained using 2-step pretraining and a ResNeXt50 autoencoder compared to 20.4% when using no pretraining. This holds true across all the backbones.

It originally appeared that a difference in random and ImageNet initialisations resulted in different performance in the end segmentations however this did not materialize in a significant manner.

5 Conclusion

2-step pretraining is an effective way of increasing the resultant accuracy of a semantic segmentation model when faced with limited labelled segmentation data. DeepLabv3 with a ResNeXt50 backbone pretrained as an autoencoder is overall the best of the tested models for this approach and did benefit greatly from the 2-step pretraining. If two separate image classification and object localisation datasets are not available pretraining on a single related dataset is still worthwhile. However can not conclusively say whether random or ImageNet initialisation or if a larger image classification or a smaller object localisation dataset are more effective for 1-step pretraining.

6 Further research

Further research is needed to examine if accuracy of the pretrained models is critical to the resultant segmentation mIOU. Self supervises and unsupervised learning approaches for pretraining should be considered as well.

References

Arnab, A.; Miksik, O.; and Torr, P. H. 2018. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 888–897.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Erhan, D.; Courville, A.; Bengio, Y.; and Vincent, P. 2010. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 201–208. JMLR Workshop and Conference Proceedings.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hong, S.; Oh, J.; Lee, H.; and Han, B. 2016. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3204–3212.

Irvin, J.; Rajpurkar, P.; Ko, M.; Yu, Y.; Ciurea-Ilcus, S.; Chute, C.; Marklund, H.; Haghgoo, B.; Ball, R.; Shpanskaya, K.; et al. 2019. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 590–597.

Lakhani, P.; Mongan, J.; Singhal, C.; Zhou, Q.; Andriole, K. P.; Auffermann, W. F.; Prasanna, P.; Pham, T.; Peterson, M.; Bergquist, P. J.; and et al. 2021. The 2021 siim-fisabio-rsna machine learning covid-19 challenge: Annotation and standard exam classification of covid-19 chest radiographs.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Zamir, A. R.; Sax, A.; Shen, W.; Guibas, L. J.; Malik, J.; and Savarese, S. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3712–3722.

Backbone	Pretraining Method	UNet				DeepLabv3	
		mIOU(%)				mIOU(%)	
		Encoder Only		Both		Encoder Only	
		Rand.	Image Net	Rand.	Image Net	Rand.	Image Net
VGG16 Classifier	2-Step	22.3±4.2	25.1±2.2	-	-	26±3.8	24.6±9
	1-Step, CheXpert	20.9±1.2	24.73±1.9	-	-	23.91±1.34	20.4±3.8
	1-Step, SFRC	19.13±9.1	21.44±5	-	-	23.41±3.7	21.63±5.1
	None	19.4±4.6	19.6±8.7	-	-	20.3±3.1	20.1±3.4
ResNet50 Classifier	2-Step	28.9±5.1	27.2±7.1	-	-	24.4±4.9	21.8±5.1
	1-Step, CheXpert	24.23±3.7	23.1±5.4	-	-	23.1±1.2	21.3±3.8
	1-Step, SFRC	25.4±3.6	24.3±3.1	-	-	20.1±6	19.6±8.1
	None	20.3±1	18.9±3	-	-	20.9±2	17.9±11.2
ResNeXt50 Classifier	2-Step	29±3.8	28.9±5.9	-	-	32.5±2	28.9±7.7
	1-Step, CheXpert	26±2.1	27±6.4	-	-	28.1±3	27.2±6.3
	1-Step, SFRC	28.7±1.3	28.2±3.1	-	-	26.4±2.4	25.2±3.7
	None	21.2±5.4	22.4±3.1	-	-	22.9±1.5	23.8±8.9
VGG16 Autoencoder	2-Step	20.31±6.49	21.6	24.5	25.7	26.1	25.8
	1-Step, CheXpert						
ResNet50 Autoencoder	2-Step	26.1±3	22.3±4.2	29.2±3.8	28.6±7.9	36.3±2.5	35.4±6.1
	1-Step, CheXpert						
ResNeXt50 Autoencoder	2-Step	27.1±6.5	23.3±2.8	37.3±2.5	37.4±7.9	38.3±5.2	39.6±5.1
	1-Step, CheXpert						

Table 3: Comparison of 2-step, 1-step and No pretraining. Random and ImageNet initialisation of network weights are split. mIOU is the average mIOU across multiple runs. ± values are the difference between the best and the worst models in that category.