* First select overall segmentation network configuration, then optimal patch size
* Maybe compare with other models
* Check keras iou implementation
* How to select model after k crossfold (select ‘winning’ model from crossfold validation and train again on all data)
* Early stopping
* “Fully Convolutional Networks for Semantic Segmentation”
* Patch extraction literature: <https://openaccess.thecvf.com/content_cvpr_2015/papers/Long_Fully_Convolutional_Networks_2015_CVPR_paper.pdf>

Some applications of deep learning methods on remote sensing can be also observed in Han et al., 2015, Yao et al., 2016, Zhang et al., 2016, Cheng et al., 2016, Cheng et al., 2017.

Patch Extraction: Patch size strongly affects the discriminative power of the network. When patches are cropped to smaller regions, we likely crop out defining features. Taking away explanatory context will render the matching more difficult. All further discussion will be with reference to the results we obtained using the patch size 256 pixels.

We trained five versions of our proposed network, each at a different patch size, in order to evaluate the effect of patch size on classification accuracy.

----------------------

Transfer Learning is the task of transferring knowledge from a source domain to a target domain. Ideally domains are similar. However, … could show that

Transfer Learning tackles the problem of overfitting by using pre-initialized weights as a starting point for model training. By using activations learned from a primary task, the assumption is that features learned can generalize to the task at hand. Transfer Learning has been shown to result in faster convergence, more robust models and better overall performance.

Transfer Learning can be done by freezing pretrained layers during model training, using features as a starting point for fine-tuning or a mixture of both. Yosinski showed that networks learn features from general to specific, assuming that features learned at earlier network stages might transfer better to improve generalization on different domain tasks.

For segmentation tasks, pretraining has been shown successful by using a pretrained classification model as encoder, cutting the fully connected classification layers and stacking them to a decoder. At this stage, the network will have learned to extract Imagenet features.

One important question with regard to transfer learning design is how similar the primary and secondary task need to be. A pretraining dataset often used is classification dataset Imagenet which contains millions of samples (describe some of the imagenet classes). By its popularity it is a great basis for feature learning and has been proven successful in many tasks. However, this dataset is mostly targeted in recognizing small everyday objects. Yosinski showed that ‘the effectiveness of feature transfer is declining as base and target tasks become less similar. But found that initializing with transferred features can improve generalization performance even after substantial fine-tuning on a new task.”

Hu and … addressed the question of pretraining with Imagenet for Remote Sensing issues. In Remote Sensing, large-scale labelled datasets are not available in the scale that Imagenet. … could interestingly show that pretraining on large (different) Imagenet results in better performance thatn pretraining on smaller (more similar) Patternnet. This exemplifies the chance that training on Imagenet bares still chances.

Hu, Yosinski and … showed that pretraining is however more successful than randomly initializing weights (training from scratch), even for remote sensing datasets. However, they used datasets that still contain objects like planes or ships, that resemble Imagenet features. Melt pond sea ice features are very homogeneouos, continuous shapes and differ from tables, chairs, regardless of the difference in perspective. Easier than complex Imagenet features. Additionally, different spectra: one-channel IR instead of three-channel RGB.

**Methodology**

**Data Preparation**

Network was trained with raw images. Conversion to image values (0,255) yielded much better performance than original temperature values (273.5,276), thus, this method was kept for all further experiments reported. (This might be due to…). Kept grayscale. Mask values were transformed to grayscale and one-hot encoded.

Preprocessing was applied network specific, which is none for Resnet34.

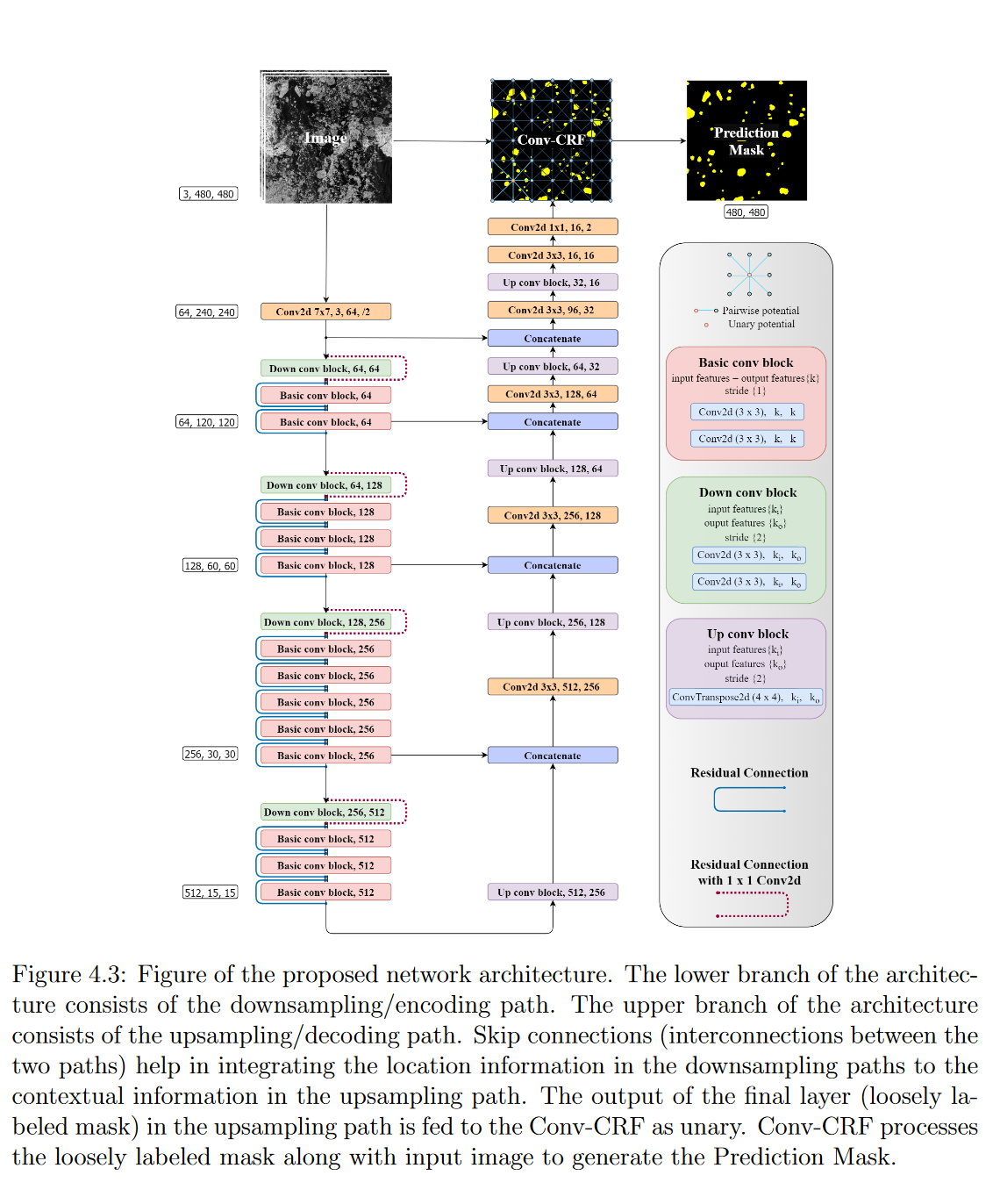
**Patch Extraction**

All images have an image size of 480, 480 after preparation. Target features melt ponds are much smaller than this image size. Reducing the image size leads to more training data, whereas receptive field is smaller. By training on different patch sizes, the required receptive field is investigated that covers sufficient context to correctly classify ponds and distinguish ocean from sea ice class. Patch extraction is a commonly used method in remote sensing classification as in…

Patch sizes of 32,64,128 and 256 have been investigated, together with entire image 480. The resulting dataset sizes and semantic content can be regarded in Table 2.

Patch Extraction is a common approach in remote sensing scenes, where image scales range to … (reference). To increase the dataset size, effect of patch extraction with different patch sizes is investigated. By extracting patches with size smaller, the receptive field is decreased while larger image size.

* Makes process faster and more efficient by splitting data into smaller chunks
* Reduces the risk of overfitting due to increased number of parameters compared with larger sets of training data. Patch sizes determine how receptive field gets affected for each filter at every layer during form extraction in CNNs.
* Larger patch size allows more contextual information from its surroundings while smaller patches can be useful if trying to extract certain features from images with less data required.
* Choosing a local region too large can results in overfitting due to limited training data leading an optimal performance not being obtained after training. Selecting patch size too small may lead to insufficiency and inaccuracy of input feature maps into deeper layers
* Smaller patches require less data but also offer fewer details about the overall image or object being studied
* Higher resolutions can improve accuracy while reducing computational time due to ability to capture more detail from larger objects.
* 256 will lead to overlapping patches 🡪 middle area will be overrepresented 🡪 however was chosen to make training set larger and don’t loose information (other methods would include mirroring)



**Model Training**

“Lack of sufficient labelled data due to tedious and time-consuming manual segmentation”. Additionally, in the current task, ice floe and melt pond shapes have high intraclass variations due to different floe and pond sizes, temperatures…

With this limited data accessible for training… To successfully train a model with small data, overfitting is a big concern. This part tests the effectiveness of different methods that tackle this problem. To allow for comparibility, a baseline model was trained and successively methods added. The baseline model and methods are described below in more detail. Models have been evaluated using 5-crossfold validation as a small test set is likely to be biased. (How exactly is crossfold implemented). The main metric considered is mean IoU which is commonly used in segmentation problems. To save training time, all models have initially been trained for 40 epochs, as first experiments showed convergence at that point already. Configurations with slightly fluctuating training curves were trained again for 100 epochs to see if performance increases. This holds especially for augmentations, as this extends the training information available.

“For testing a model, patches of size K were extracted from the test image using a stride of K and segmentation was performed on each patch and the results were stitched back to get the final result.”

Training is performed with cross entropy loss function

Modified U-Net

* Batch normalization layers
* Upsampling instead of transposed
* Resnet34 backbone
* Input image size is fixed for each model used
* Architecture contains approximately … parameters
* Stride… kernel…
* A summary of the architecture can be found in supplementary materials
* As the original implementation was built for RGB images, grayscale images have been stacked as common method reported in …

**Baseline Model**

* Optimizer is set to Adam which has an adaptive learning rate…
* 0.2 was used as test ratio, resulting in 25 images for training and … for testing

**Transfer Learning**

* Imagenet weights were used
* ResNet34 as backbone as has been successfully done by … ResNet34 is a fully connected classification model that uses skip connections to tackle the problem of vanishing gradients. By splitting the network at the last convolutional layer, it can be stacked before decoder. To integrate skip connections as in U-Net, activations are transferred at the relevant stages.
* Model starts with Imagenet weights and are fine-tuned during model training. In the baseline model, all layers are set to trainable.
* For training from scratch, the same model architecture has been used but without loading pretrained weights
* Referring to … that showed that half freeze half finetune results in better performance, in an ablation study, half of the layers were frozen
* “train a base network and then copy its first n layers to the first n layers of a target network. The remaining layers of the target network are then randomly initialized and trained towards the target task. One can choose to backpropagate the errors from the new task into the base (copied) features to fine-tune them to the new task, or the transferred feature layers can be left frozen, meaning that they do not change during training on the new task. The choice of whether or not to fine-tune the first n layers of the target network depends on the size of the target dataset and the number of parameters in the first n layers. If the target dataset is small and the number of parameters is large, fine-tuning may result in overfitting, so the features are often left frozen. On the other hand, if the target dataset is large or the number of parameters is small, so that overfitting is not a problem, then the base features can be fine-tuned to the new task to improve performance. Of course, if the target dataset is very large, there would be little need to transfer because the lower level filters could just be learned from scratch on the target dataset.“
* “For our experiments, we use the CNN based on VGG19. VGG19 is… To adapt the CNN-VGG19 architecture to an FCN some modifications are required: The final classification layer is discarded and replaced with a 1x1 convolution and with the channel dimension of the number of used classes. Deconvolutional layers are introduced for bilinear upsampling of the coarse outputs to pixel-dense outputs, using upsampling (not transposed)
* Skip connections between encoder and decoder: “Fusing fine layers and coarse layers lets the model make local predictions that respect a global structure. The FCN fuses the upsampled output of the VGG19 with predictions computed on top of the third and fourth pooling layer.”
* “for both transfer learning experiments all trainable variables of the FCN are available during backpropagation to ensure adapting all parameters for the different resolutions and image sensing methods of the RS data”
* “4-fold cross validation: each scene is split into four equal data strips. Oout of the four data strips, three strips are used as training samples which are randomly shuffled after each epoch and the remaining strip is used for validation. Process is repeated four times with each of the four strips used exactly once for validation.”
* Transfer learned for how many epochs

**Evaluation Metrics**

Quantitative Segmentation Metrics:

1. Pixel Accuracy
2. Mean Intersection over Union (mIoU)
3. Per class IoU, as class frequencies are imbalanced
4. F1
5. (fw frequency weighted iou)

Melt Pond fraction (median of MAE over test set to be robust against outliers)

* Maybe mean IoU of unsupervised VIS and supervised

Qualitative Metrics: Compare 10 images (somehow subjective). “Additional results are in the supplementary material”. In case of smaller ice floes, water is misclassified as melt ponds several times…

**Results**

In this section, the capabilities of deep learning for slum mapping in different [remotely sensed data](https://www.sciencedirect.com/topics/computer-science/remotely-sensed-data) sets with varying characteristics are analyzed subject to the quantitative results of the performed semantic segmentation experiments.

**Discussion**

* Test set might be too similar to training set
* Other optimizer might be good to use
* Observe the influence of learning rate and batch size. Generally, smaller batch size should have a smaller learning rate.
* Using very limited labeled samples and large number of unlabeled pixels, semisupervised methods have presented good performance (Tuia and Camps-Valls, 2009, Yang et al., 2014, Wang et al., 2015, Huo et al., 2015, Wan et al., 2015, Romaszewski et al., 2016, Ma et al., 2016)

**Transfer Learning: How transferable are features in deep neural networks?** (Yosinski) – Learning many related tasks at the same time with backpropagation (Caruana; 1995) – Deep lEarning of Representations for Unsupervised and Transfer Learning (Scotland 2012)

Yin (2017): Fine-tuning and visualization of CNN; Amsterdam

* The primary task (Imagenet, composed of natural images) is not very similar to the secondary task (RS). The number of channels differ, images are from different spectra

<https://ieeexplore.ieee.org/document/1495508> (patch size)

This “whole-image” approach has two deficiencies. First, there are considerable differences between the numbers of pixels belonging to each category. This may cause the infrequent categories to be simply ignored by the learning process. Second, processing a whole image at once can be seen as being equivalent to processing a large number of 40 × 40 pixel windows in a batch. Previous studies have shown that performing a weight update after each sample leads to faster convergence than updating the weights after accumulating gradients over a batch of samples [20]. Therefore, we chose to break up the training images into a series of overlapping 40 × 40 windows that can be processed individually. Overall, from the 50 frames in the training set, 190; 440 windows of size 40 × 40 pixels were extracted. To each such window was associated the desired labels (for M1 and M2) of the central pixel in the window. Each pair of window and label was used as a separate training sample for the convolutional network, which, therefore, produced a single output vector (a one pixel output map). There were wide variations in the number of training samples for each category: 3333 windows were labeled nucleus, 12 939 nuclear membrane, 80 142 cytoplasm, 39 612 cell wall, and 54 414 external medium. To correct these wide variations, a class frequency equalization method was used. A full learning epoch through the training set consisted in 272070=5×54414 pattern presentations. During one epoch, each sample labeled “external medium” was seen once, while samples from the other categories were repeated 54414/P times, where P is the number of samples from that category. Therefore, each category was presented an equal number of times (54414) during each epoch.

<https://ieeexplore.ieee.org/abstract/document/8082108> (for autoencoder pretraining)

* Our network architecture is based on the so-called encoder-decoder paradigm. The input is first transformed into a typically lower dimensional space via a conv subnetwork and then expanded to reproduce the initial data by a deconv subnetwork (decoder).
* Moreover, the trained unsupervised Conv–Deconv network can be adapted to the classification of hyperspectral data by cutting off the deconvolutional subnetwork, replacing the loss function, and fine-tuning it to the new task, i.e., adjusting the weights using backpropagation. With this approach, typically much smaller training sets are sufficient.

<https://ieeexplore.ieee.org/abstract/document/8314449>

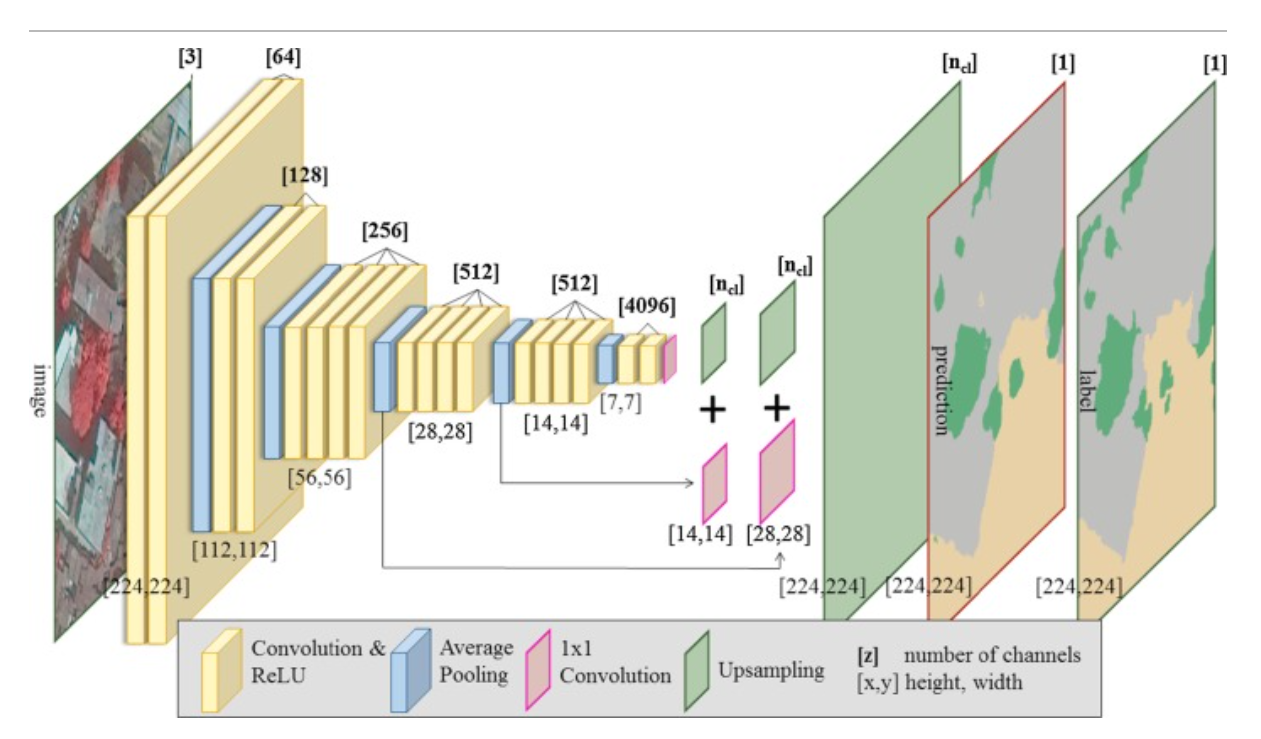
* Optical images reflect the chemical characteristics of the scene and follow a perspective imaging geometry
* Design a pseudo-siamese network architecture with two separate, yet identical convolutional streams for processing SAR and optical patches in parallel instead of weight shared simaese network in order to deal with the heterogeneous nature of the input img

<https://openaccess.thecvf.com/content_CVPR_2019/papers/Sun_Not_All_Areas_Are_Equal_Transfer_Learning_for_Semantic_Segmentation_CVPR_2019_paper.pdf> (TL more advanced)

* Recent works have been trying to apply GANs for domain alignment
* Transfer knowledge between source and target domain more effectively

<https://www.sciencedirect.com/science/article/pii/S0924271619300383> (TL in semantic segmentation)

* Deep learning algorithms attempt to automatically learn multiple levels of representations exclusively from its input data, without the need of additional user input (Zhu et al 2017) 🡪 Effectiveness + task of training and prediction is facilitated
* Deep learning can learn more abstract and discriminative semantic features with growing depth. Training of neural networks, is usually performed using pretrained networks on large image datasets, e.g., COCO ([Lin et al., 2014](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b9000)), Pascal VOC ([Everingham et al., 2010](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b9005)) or ImageNet ([Deng et al., 2009](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0030)) which in general reach impressive accuracies ([Hu et al., 2015](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0055), [Zou et al., 2015](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0285)).
* Inductive transfer learning enables to further improve the learning task where [backpropagation](https://www.sciencedirect.com/topics/computer-science/backpropagation) successfully re-weights labeled data from natural image datasets, e.g. ImageNet to solve new problems in remote sensing datasets (e.g. [Maggiori et al., 2017](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0130), [Marmanis et al., 2016](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0140), [Nogueira et al., 2017](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0155), [Kang et al., 2018](https://www.sciencedirect.com/science/article/pii/S0924271619300383" \l "b0080))
* Observed that trasnfer learning was unsuccessful from optical to SAR representation



<https://arxiv.org/pdf/1411.1792.pdf> (Yosinski)

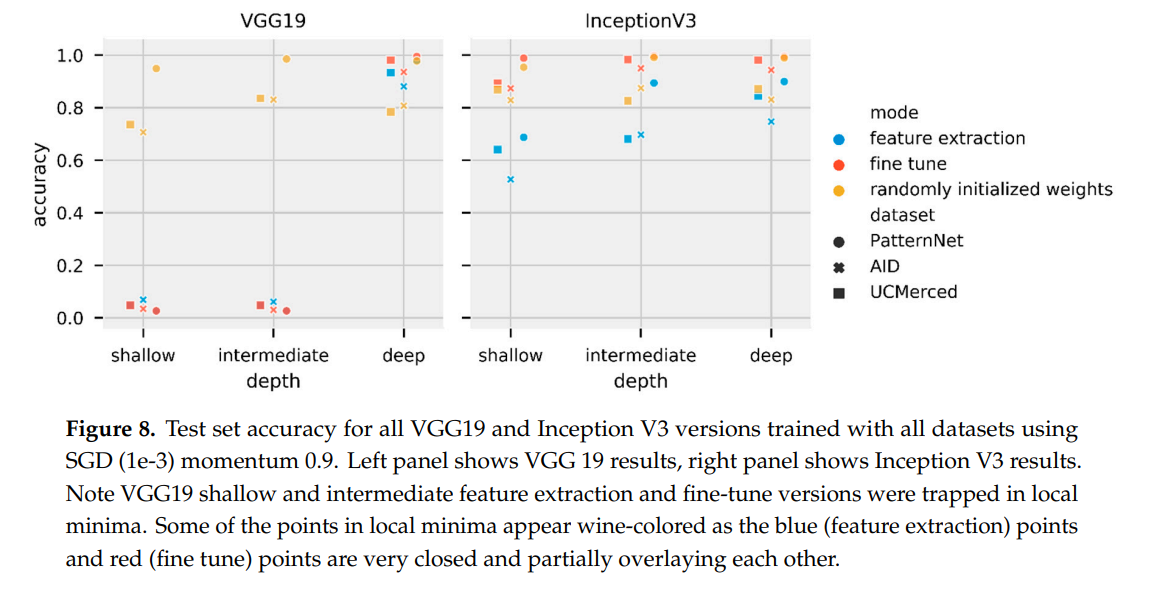
* the specialization of higher layer neurons to their original task at the expense of performance on the target task
* investigate influence of transferring features from bottom, middle or top of the network
* transferability of features decreases as the distance between the base task and target task increases, but transferring features even from distanct tasks can be better than using random features
* initializing a network with transferred features from almost any number of layers can produce a boost to generalization that lingers even after fine-tuning to the target dataset
* literature transfer learning (Caruana 1995; Bengio 2011; Bengio 2011)
* “In transfer learning, we first train a base network on a base dataset and task and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specifc to the base task”
* “When the target dataset is significantly smaller than the base dataset, transfer learning can be a powerful tool to enable training a large target network without overfitting; Recent studies have taken advantage of this fact to obtain state-of-the-art results when transferring from higher layers (Donahue et al., 2013a; Zeiler and Fergus, 2013; Sermanet et al., 2014), collectively suggesting that these layers of neural networks do indeed compute features that are fairly general. These results further emphasize the importance of studying the exact nature and extent of this generality.”

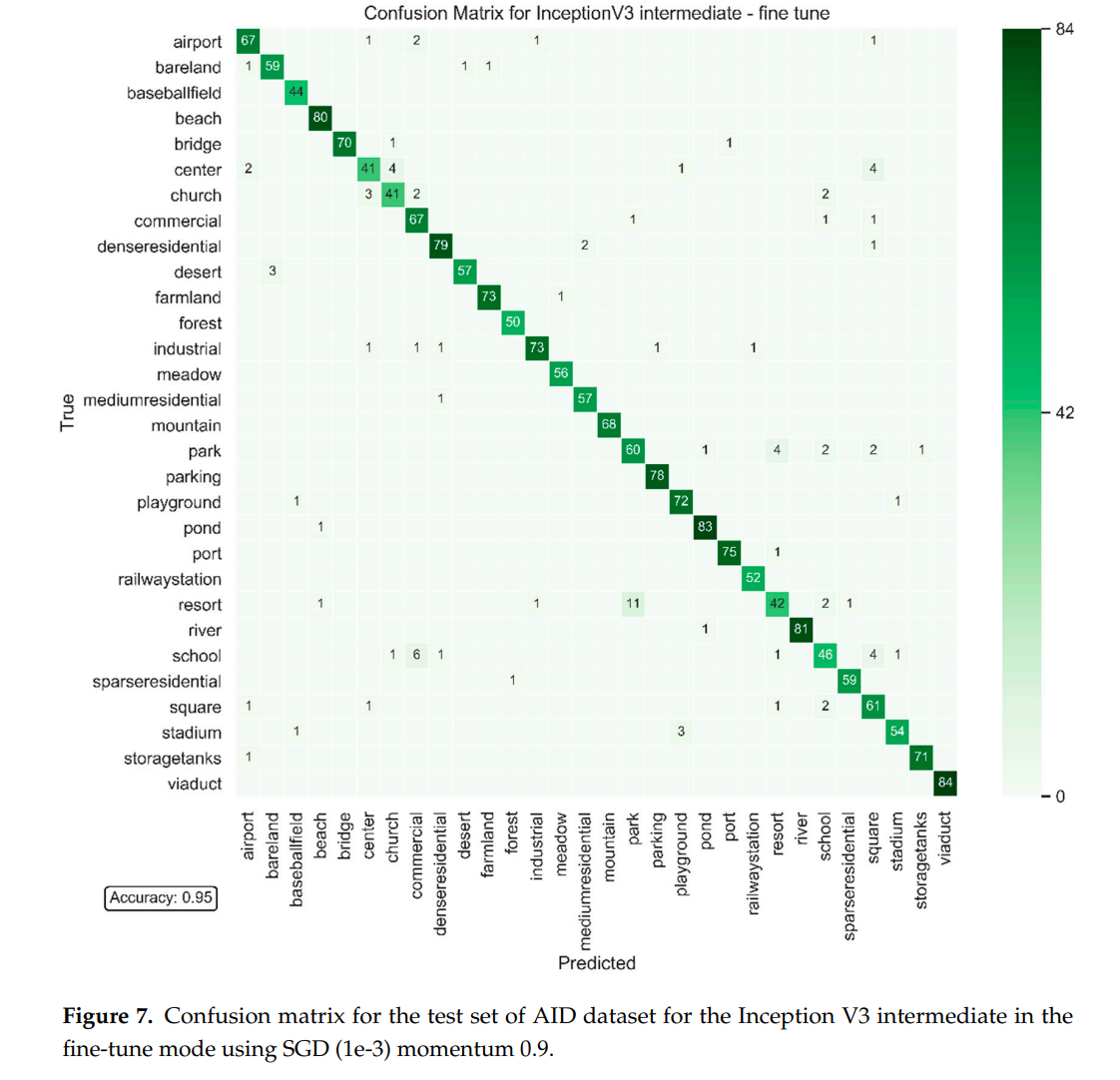
<file:///C:/Users/marle/Downloads/remotesensing-07-14680.pdf> (Hu 2015):

* Features from pre-trained CNNs generalize wel to HRRS datasets
* DL methods (21-23) have achieved great success not only in classic problems… These methods have achieved dramatic improvements beyond the state-of-the-art records in such broad domains, and they have attracted considerably interest in both the academic and industrial communities (22).
* In general, deep learning algorithms attempt to learn hierarchical features, corresponding to different levels of abstraction. The deep convolutional neural networks (CNNs) [24], which are acknowledged as the most successful and widely used deep learning approach, are now the dominant methods in the majority of recognition and detection tasks due to the remarkable results on a number of benchmarks [25–28]. CNN is a biologically inspired multi-stage architecture composed of convolutional, pooling and fully-connected layers, and it can be efficiently trained in a completely supervised manner. However, it is difficult to train a high-powered deep CNN with small datasets in practice. At present, many recent works [29–34] have demonstrated that the intermediate activations learned with deep CNNs pre-trained on large datasets such as ImageNet [35] can be transferable to many other recognition tasks with limited training data.
* Considerable popularity of CNN in the CV community (25,27,28,32,33,42)
* However, it is very difficult to train a deep CNN which typically contains millions of parameters for some specific tasks, with a small number of training samples. Increasingly more works have shown that intermediate features extracted from dCNNs that are trained on sufficiently large-scale datasets such as ImageNet can be successfully applied to a wide range of visual recognition tasks e.g. scene classification (29,34), object detection (34,44), image retrieval (25,43)
* (also explain architecture of classical CNN if needed)
* **36, 37** related work (transfer learning for HRRS)
* In the remote sensing field, there is still a lack of investigations on using CNNs for HRRS scene classification; hence, we attempt to present a comprehensive study on this topic. Our work is most related to [36,37], which are concurrent works with our own. In [37], the authors employed pre-trained CNNs and fine-tuned them on the scene datasets, showing impressive classification performance, whereas we transfer the pre-trained CNNs for scene datasets without any training modalities (fine-tuning or training from scratch) and simply take the pre-trained CNNs as fixed feature extractors. In [36], the authors evaluated the generalization power of CNN features from fully-connected layers in remote sensing image classification and showed state-of-the-art results on a public HRRS scene dataset. In contrast to [36], we investigate CNN features not only from fully-connected layers but also from convolutional layers, we evaluate more pre-trained CNN models, and more detailed comparative experiments are presented under various settings. Thus, this paper provides a more systematic study on utilizing pre-trained CNN for HRRS scene classification tasks.
* 39!!!

<https://www.mdpi.com/2072-4292/12/1/86> (Convolutional Neural Network for RS Scene Classification: Transfer Learning Analysis)

* “we find transfer learning from models trained on larger, more generic natural images datasets outperformed transfer learning from models trained directly on smaller remotely sensed datasets”
* Conventional scene classification techniques rely on low-level visual features to represent the images of interest. Such low-level features can be global or local. Global features are extracted from the entire remote-sensing image, such as color features, texture features and shape features. Local features are extracted from image patches that are centered about a point of interest. However, these global and local features are handcrafted, time-consuming, needs heuristic or ad hoc design decisions.
* Hu et al: the more representative and higher-level features, which are abstractions of the lower level features, are desirable and play a dominant role in scene classification task. The extraction of high-level features promises to be one of the main advantages of deep learning models.
* “Despite CNN’s powerful feature extraction capabilities, Hu et al found that in practice it is difficult to train CNNs with small datasets”
* Yosinski anf Yin observed that parameters learned by the layers in many CNN models trained on images exhibit a very common behaviour. The layers closer to the input data tend to learn general features, resulting in convolutional operators akin to edge detection filters, smoothing, or color filters. Then there is a transition to features more specific to the dataset on which the model is trained
* “In Transfer Learning, the filters learned by a CNN model on a primary task are applied to an unrelated secondary task. The primary model can be used as a feature extractor or as a starting point for a secondary model.”
* Despite the success of transfer learning in applications in which the secondary task is significantly different from the primary task (<https://www.nature.com/articles/nature21056>, <https://library.seg.org/doi/10.1190/segam2018-2998567.1> , <https://library.seg.org/doi/abs/10.1190/segam2019-3215401.1> ), the remark that the effectiveness of TL is expected to decline as the primary and secondary tasks become less similar (Yosinski) is commonly made. Although Yosinski concluded that using TL from distant tasks perform better than training CNN from scratch, it remains unclear how the amount of data or the model used can influence the performance.
* TL from natural images to RS is possible: Despite the relatively large difference between primary and secondary tasks, TL generally outperformed scratch. Fine-tuning models primarily trained on Imagenet outperformed finetuning on PatternNet, reason might be that Imagenet is complex dataset where intraclass variance (
* It is easier to overfit models with many weights
* When initial layer
* The backbone, when containing the weights learned during training for the primary task, will have its layers presenting the transition from general to specific features
* Explanation of SGD and why it is better than full gradient descent
* Three transfer learning modes: Feature extraction (freeze pre-trained layers), fine tuning (starts as feature extraction for half of the epochs but eventually allows all layers of the model to learn for half of the epochs), randomly initialized weights mode starts the entire model with randomly initialized weights after which all the weights are updated during training
* Validation and test set metrics better than training set metrics can be caused by the dropout layer, as during training less information is available for the model, or simply because of the data split; the training set is generally larger than validation and test sets and can incorporate a higher complexity in its samples
* Unlike poor generalization performance of adaptive methods compared to SGD optimizers reported by Wilson (65), results do not find significant differences in performance for optimizers tested. SGDs had slightly worse performance.
* Confusion matrix
* Provide information about training time (The executrion times are provided as simple general reference and lack more detailed analysis of performance – the computer was not entirely dedicated to experiments, thus speed might have been affected)
* In table, show the best performing training for each mode
* Adamax results in transfer learning in general better than SGD, step size 2e-3 too large for fine tune VGG19
* Difference in datasets: feature extraction mode limited. In fine-tuning overfitting reduces when layers are set to trainable.
* “Our results align with their findings”
* “all of the loss and accuracy per epoch can be accessed in the supplemental materials)
* Despite feature extraction limitations, the results show that transfer learning is an effective deep-learning approach that should not be discarded if the secondary task is not too similar to the primary task





<https://www.mdpi.com/journal/remotesensing/special_issues/DeepTransfer_Learning> (Special Issue)

<https://ieeexplore.ieee.org/document/8809071/references#references> (what data are needed for semantic segmentation in EO) few citations!

* Pretrained converges faster and better accuracy than from scratch

<https://ieeexplore.ieee.org/document/7301382/citations#citations> (Do deep features…)

* Evaluation of generalization power of pretrained ConvNets from everyday objects to the aerial and remote sensing domain
* “ConvNets have shown astounding results even in datasets with different characteristics from which they were trained, feeding the theory that deep features are able to generalize from one dataset to another.”
* “Although ConvNets were not the best descriptors for the coffee dataset, they could still perform well. This is interesting specially because the ConvNets used here were trained to recognize objects which is a very different scenario in relation to recognizing coffee regions. This also shows the generalization power of ConvNets”

<https://arxiv.org/pdf/1411.1792.pdf> (Yosinski: How transferable are features in deep neural networks?)

<https://ieeexplore.ieee.org/abstract/document/9122009>

used