IOSR Journal of Applied Geology and Geophysics (IOSR-JAGG)
e-ISSN: 2321–0990, p-ISSN: 2321–0982.Volume 8, Issue 5 Ser. I (Sep. – Oct. 2020), PP 24-36
www.iosrjournals.org

Automatic First Break picking with Deep Learning

Chiel Fernhout[1], Paul Zwartjes[2], Jewoo Yoo[2]
[1](formerly Delft University of Technology, The Netherlands)
[2](Aramco Overseas Company BV, The Netherlands)

Abstract:
A key step in seismic data processing is first break (FB) picking, or rather, determining the onset of the first seismic arrivals in seismic records. FB picking is tedious and time-consumingtask and robustness and efficient automatic method are essential. Many automated FB-picking algorithms already exist that reduce the dependence on human interaction. The goal of this project is to improve automated FB picking by capturing the expertise of FB picking through supervised training of deep neural network from the field of machine vision. We have evaluated several neural network architectures and found a seven layer U-net with skip connections to provide the best results on seismic shot gathers portioned into 64x64 windows. We applied the trained networks to unseen data with encouraging results, provided pre-processing is applied to make the unseen data similar to the training data. Additionally, by adding a small number of picked gathers from a new dataset to an existing training dataset, the algorithm performs well on the new dataset. We also found that the network is robust against a small amount of mispicks in the training data, and can actually improve these erroneous picks.
Key Word*:seismic processing; first break; deep learning; U-net*

Date of Submission: 20-09-2020 Date of Acceptance: 05-10-2020

I. Introduction

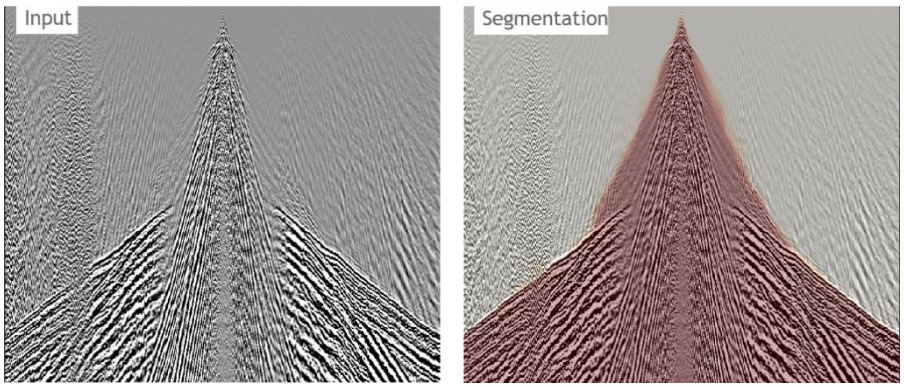

Figure no 1: Definition ofthe **s**egmentation objective for seismic shot gathers. The red colored segment on the right resembles all the source-generated energy. The other segment contains just recordings of ambient noise. The boundary between the two segments is called the "First Break" and resembles the first arrival of seismic energy.

Seismic data is essential to obtain a structural image of the subsurface and seismic data processing essential to obtain that image. An important topic in seismic data processing is first break picking, where the first break (FB) is the first recorded source generated seismic event. The FB is characterized by the arrival time of the seismic event and the distance (offset) between the source and the receiver. Human vision applied to FB picking still beats the best algorithms in terms of quality, yet is not capable to handle the quantity of data, therefore an efficient FB picking algorithm is essential. Automated FB-picking methods generally are divided into three categories (Yoo et al., 2019): (1) cross-correlation methods, (2) energy/amplitude-based methods, and (3) methods based on neural network. Most methods are single-trace based. This single-trace approach has computational advantages, but typically struggles on noisy traces. These conventional FB picking methods require manual parameterization and fine-tuning to avoid mispicks.Some single-trace methods have post-processing steps, such as median filtering, to address erroneous picks. Therefore, there is a need for an easy to use and robust first-break picker that can deal with relatively low signal-to-noise ratio.

Can we capture the essence of human vision into a machine vision based algorithm?The answer is clearly yes and recently, a variety of neural network architectures, based on convolutional neural networks, have been presented recently (Xie et al., 2018; Tsai et al., 2018; Duan et al., 2018; Yuan et al., 2018; Liu et al., 2018).In this paper, we address and compare these architectures.

The problem of first-break picking with deep neural networks can be approached in two ways; as a regression or a classification problem. When the input is a 2D gather and we have a model that outputs a 1D array with the FB arrival times of each trace, the model performs a regression task. In classification, we can assign a label to an entire image or to each individual sample in the image, in which case the proper term is segmentation. Here we assign a label to each sampleto indicate whether the sample is below or above the FB. In this paper, we use the segmentation.Segmentation neural networks need to be trained via supervised learning, which means we need to have first breaks and derive the segments (a.k.a. masks) from them. We use FB-picking method presented by Yoo et al. (2019) to generate these reference picks. Figure no 1 shows how a first break pick is convertedinto a binary mask. The seismic data below the first break corresponds with this mask. Since we use this FB-picking method to generate training data, we assume that the FB picks are correct. During the training, the models learns to generalize and small amounts of errors in the training FB picks do not matter.

II. Method

Artificial Neural Network (ANN)

The workhorse of deep learning is the Artificial Neural Network (ANN). In a supervised learning process, an ANN can learn to predict a desired result from data. The parameters in the neural network are the so-called weights and biases. During training, these parameters are updated via a gradient descent algorithm, which calculates the derivative of the loss function. The stochastic gradient descent (SGD) algorithm is a gradient descent algorithm that operates on one example at a time. This leads to very erratic updates and wild swings in the loss function. A more practical approach is a gradient descent algorithm that operates on batches of examples at a time. The size of these batches are determinedmainly by the memory available, the size of each training example and the size of the ANN.We have used the Adam gradient descent implementation (Kingma and Ba, 2015).A pitfall in training is overfitting, which occurs when the model performs significantly better on the training data than on test data, and usually means involves fitting noise instead of data. When a model has been overfit, the model is not able to generalize what it has learned to unseen data. Artificial Neural Networks (ANNs) are prone to overfitting due to their many parameters and nonlinearities. Common strategies to prevent overfitting are regularization, dropout (Srivastava et al. 2014) and early stopping.

The specific make-up of the neural network is called the architecture. Designing a good architecture is something of a black art. There a few clear and concise rules. Instead, over time, certain architectural design choice have proven their robust and are preferred. In the following sections, we will discuss some aspect of neural network architectures.

In the ANN, all nodes of a layer are connected to all the nodes of the previous and next layers, and the layers in an ANN can be viewed as a succession of matrix multiplications. The main drawback of fully connected ANNs is that they have many parameters to train (the matrix elements a.k.a. weights). Lecun et al. (1998) showed that an increase of input size exponentially increases the number of weights to be trained, making fully connected ANNs impractical. The authors also noted two other drawbacks of fully connected networks: for recognition of images, the network has no built-in invariance with respect to translations or local input distortions. This can be achieved by making the network more complex, exaggerating the problem stated above. Another drawback is that a fully connected architecture disregards the input topology completely. Images have a strong local structure in 2D, which contains useful information for classification. Both problems can be overcomewith a Convolutional Neural Network (CNN).

Convolutional Neural Network

The Convolutional Neural Network (CNN)mimics human visual data processing, because it consists of layers that are sensitive of features at different scales. The building blocks of CNNs are its filters, grouped in convolutional layers, and the parameters of the CNN are its filter coefficients. While the filters of the network are learnedvia gradient descent, they extract basic information or features from the data. The size of a filter is called its receptive field, in the analogy to the human eye. The invariance that the CNN architecture incorporates is achieved by forcing the extractions of local features by restricting the receptive fields to be local, by imposing constraints on the weights, and by spatial or temporal sub-sampling. These three architectural ideas combined in a CNN are the reason that the network is in some degree shift, scale, and distortion invariant. By convolving an image with a filter, the output will tell us how much and where the image is similar to the filter (therefore, it is mathematically a correlation).

Combining simple filters results in the recognition of more complex structures by successive layers. In addition, the deeper layers are able to see larger objects. The Effective Receptive Field (ERF) of a specific layer

deeper in the network can see much more than only its 3 x 3 filter. Le and Borji (2017) have provided formula to calculate the ERF. For a network with two downsampling layers and six convolutional layers with 3x3 filters, the ERF of the deepest layer is 32. This means that a single sample in the last layer has seen information from an area of 32x32 in the input layer. Downsampling layers significantly increase the ERF.

Encoder-Decoder

Forcing the network through a bottleneck, first a dimension reduction, followed by a dimension increase, improves performance. This bottleneck forces the network to learn a sparse representation of the data, while at the same time learning how to reconstruct it. Such an encoder-decoder network is also called an Auto-Encoder and is often used to remove noise. The noise removal also helps to prevent overfitting.The contracting path between the input and the sparse representation is called the encoder. The reconstruction path between the sparse representation and the output is called the decoder.

A downsampling layer (a.k.a. pooling layer) is used to perform a dimensionality reduction (aggregation). The inspiration for the downsampling layer comes from a model of the visual cortex in mammals, which contains simple and complex cells. The simple cells perform a feature extraction, while the complex cells combine features to a more meaningful whole. An upsampling layer is used to increase the size of the data. There are several methods available to upsamplesuch as 'transpose convolution', 'upsampling' and 'max-unpooling'.

Residual Blocks and Skip connections

In a standard architecture, the layers are connectedin a sequence, where the output of one layer is the input for the next. A skip connection is when a layer is connected to two or more layers deeper in the network, instead of to the next layer. The ResNet architecture (He et al., 2016)is such an architecture that cleverly uses skip connections to improve performance of deep CNNs. He'sResNet architecture groups two convolutional layers and adds a skip connection that bypasses the two convolutional layers (Figure no 2). Effectively, when the two convolutional layers do not improve the performance of the network, they can be bypassed and the network continues to function properly. A residual block is considered one layer because the block of two convolutional layers can be skipped. Residual blocks help to avoid the vanishing gradient problem, as during backpropagation of the gradient, more of the gradient reaches the earlier layers in the network.

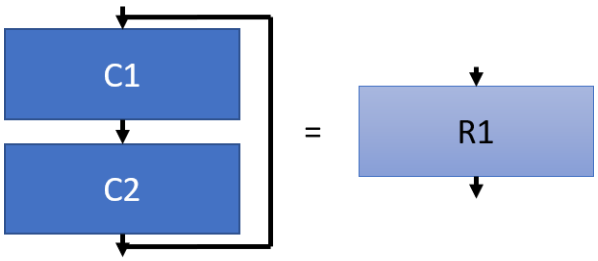

Figure no 2: The skip connections in a ResNet architecture. The blue blocks are conventional convolutional layers in a neural network. The light blue box consists of two convolutional layer and a 'skip connection', which combines the input to the first convolutional layer with the output of the second layer.

Image segmentation

For a CNN it is difficult to simultaneously characterize and locate an object precisely. The first couple of layers are good in detecting boundaries of objects with precision, but are less sure about classifying these patterns. Deeper layers in the network combine classifications and localizations in a nonlinear way (Long et al., 2015). Deeper layers have a larger receptive field due to a stack of filters from the previous convolutional and downsampling layers. The latter makes the input coarser. The resolution is smaller, thus the location is less precise, but the classification of the patterns are more confident. This is a problem because both need to be optimized for a well-performing segmentation.

Neural Network Architectures for First Break segmentation

Several different architectures for image segmentation have been proposed, such as the SegNet (Badrinarayanan et al., 2017), the Fully Convolution Net (Long et al., 2015) and the U-net (Ronneberger et al., 2015).

U-net: The U-netwas developed in the field of biomedical imaging (Ronneberger et al., 2015). The U-net, presented inFigure no 3, has a symmetrical encoder-decoder, meaning that there are also convolutional layers betweenthe upsampling layers in the decoder. These convolutional layers help the decoder to learn to improve the upsampling. The grey arrow in Figure no 3 indicates skip connections from the encoderto the decoder. After

every upsampling operation, the accuracy (mostly) of the boundaries is enhanced by adding information from theencoder.

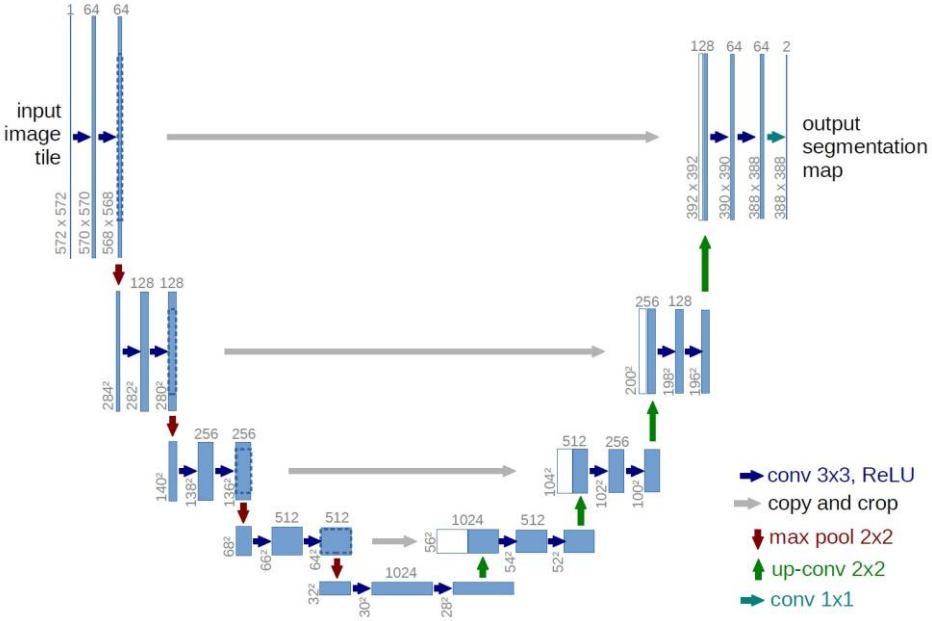

Figure no 3: The U-net architecture as presented in the original paper (Ronneberger et al., 2015). Up-convolution is a form of upsampling. Each blue box corresponds to a multi-channel feature map. The number of filters is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

SegNet: Badrinarayanan et al. (2017) proposed a fully convolutional architecture for image segmentation called SegNet (Figure no 4). Like the U-net, the SegNet has a symmetrical encoder-decoder. The goal was to create an efficient architecture in terms of memory and computation time. The SegNet uses a special type of upsampling layer called the (max-) unpooling layer. According to the authors: ”Reusing the max-pool indices in the decoding process has several practical advantages; (i) it improves the boundary delineation, (ii) it reduces the number of parameters enabling end-to-end training, and (iii) this form of upsampling can be incorporated into any encoder-decoder architecture ... with only a little modification”. Since the boundary is one of the most important aspects for First Break (FB) picking, this type of architecture could be of great value. The authors of the SegNet remark that this form of upsampling can be incorporated into any encoder-decoder architecture.

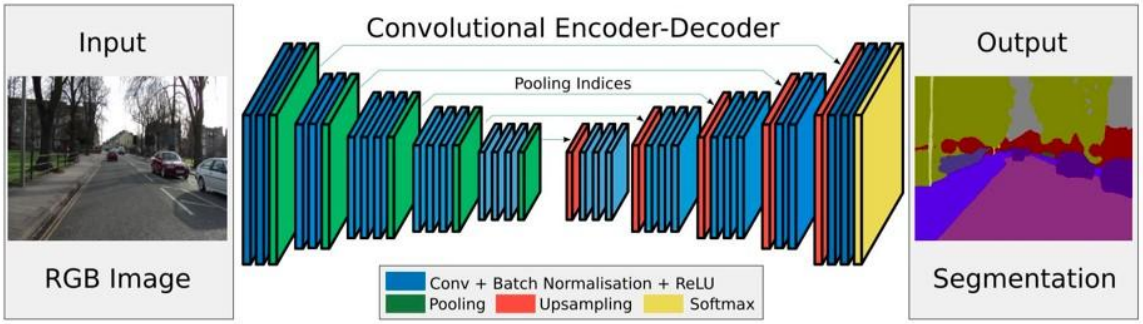

Figure no 4: SegNet architecture presented in the original paper. Source: Badrinarayanan et al. (2017)

Feature maps and filter response

To understand why the models work, we look at how the neural network transforms the input data with a series of filters. To demonstrate this, we take the U-net architecture. In the hidden layers, the Rectified Linear Unit (ReLU) activation function is used. The ReLu function puts to zero negative values and leaves positive values as is. The input data is convolved with (learnable) 3x3 filters and the output is called a feature map. In the first layer we have 64 3x3 filters, so we also get 64 feature maps. **Figure no 5**, **Figure no 6** and **Figure no 7** show the results input data and feature maps from a number of filters from different layers. It is clear that the different filters detect different patterns, triggering or pre- or post-FB events, highlighting or ignoring noise or being useless.

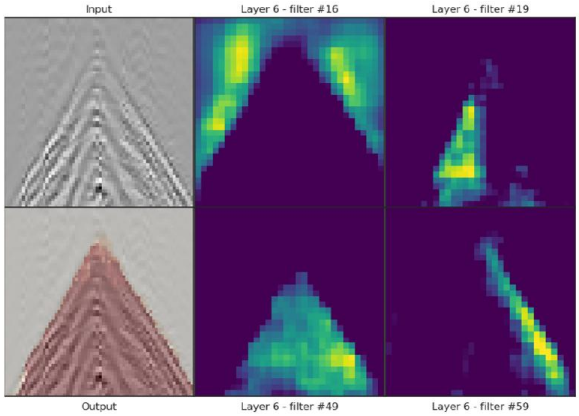

Figure no 5:A close-up of a gather from the Poland data. Four feature maps generated by the sixth convolutional layer, in the decoder part, of a U-net are visualized. Different filters focus on different structures. Filter 16 is activated for structures above the first break. Filter 49 is activated for the structures after the first break. Filter 19 highlights the negative dips, whereas filter 59 is activated for positive dips near the first break.

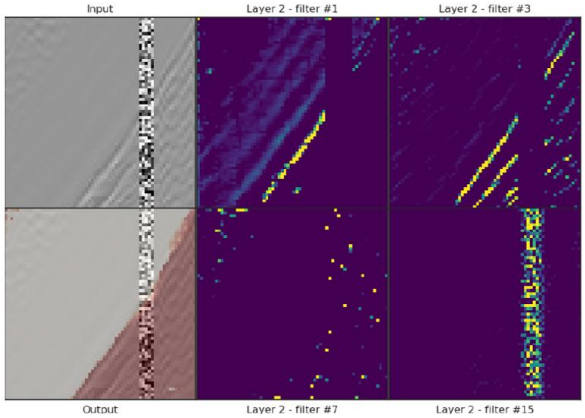

Figure no 6: As Figure no 5, but with feature, maps visualized after the second convolutional layer for a gather with noise added. Filter 1(top middle) triggers on pre-FB events, whereas filter 3 (top right) highlights post-FB events. Both filters ignore the noisy traces. Filter 7 (bottom middle) does not appear useful. Filter 15 (bottom right) only sees the noisy traces.

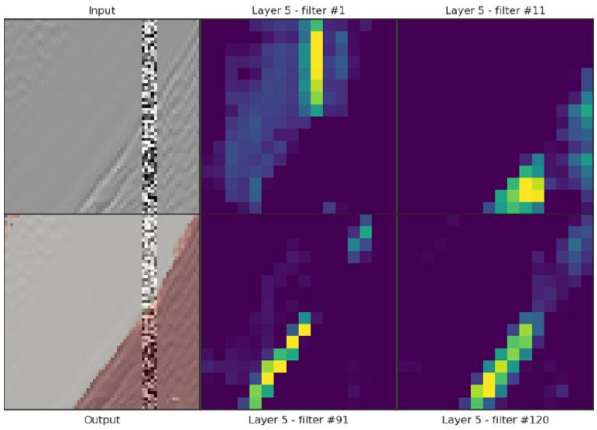

Figure no 7: As Figure no 6, but with feature maps from layer five visualized in the center of the network, when it has gone through two max-pooling layers. The feature map by filter 91 (bottom middle) has the noisy traces removed, while the filter 1 map has the noise enhanced (top middle). The feature maps from filter 120 (bottom right) and 11 (top right) have interpolated through the noise.

Grid search over architectures

To have a systematical procedure to validate which architecture is consistently better for segmenting seismic data, we ran a grid search. The following six architectures were tested: (1) SegNet, (2) U-net, (3)

ResNet, (4) ResNet+SegNet (RS), (5) ResNet+U-net (RU) and (6) ResNet+SegNet+U-net (RSU). We also tested all these architectures on all possible combinations of the three data sets and varied the neural network depth from 1, 2, 4, to 7 layers. For example, a four-layer U-net with ResNet block (RU) architecture is a U-net with four layers in the encoder, four in the decoder and has skip connections added. Figure no 8 shows the architectures.

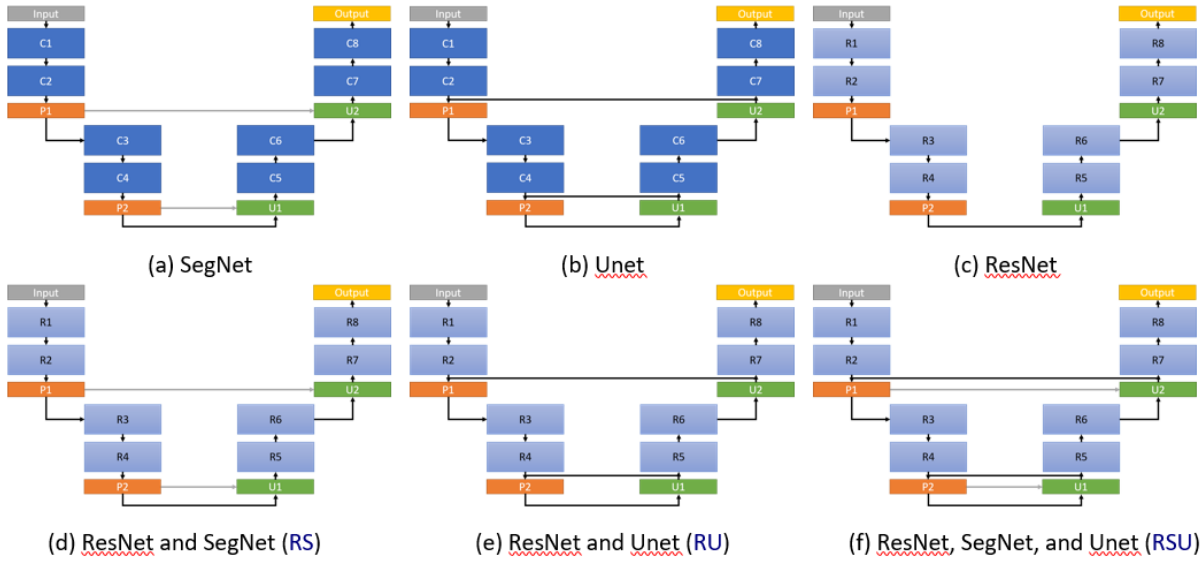

Figure no 8: Examples of two-layer architectures evaluated. Orange blocks are downsampling layers, green blocks upsampling layers. Dark blue blocks are the convolutional layers and the light-blue blocks are the residual blocks which we count as one layer as presented in Figure no 2. Dark arrows represent connection between the layers, while the light arrows represent the transfer of indices.

Algorithm

As mentioned previously, we divide the seismic shot gathers into two segments: pre-FB, denoted by zeros, and post-FB, denoted by ones. The post-FB segment contains source-generatedevents. The pre-FB segment contains noise, and events by external sources or even other seismic shots.The boundary between the two classes is the first break, by definition. However, the accuracy of the first break itself is not directly measured as a performance metric during training.The classification accuracy is measuredby the (negative) log loss metric, with values from infinity to zero, where zero is a perfect score. The log-loss function is defined by:

$$\mathcal{L}(\mathbf{W},\mathbf{y},\mathbf{x},) = \sum_{i} y_i ln\, f(\mathbf{W}; x_i) + (1 - y_i)\, ln\, f(\mathbf{W}; x_i)$$

where **W** is the matrix containing all the weights of the model, **y** are the true segment values derived from the reference FB pick, **x**is the input shot gather, and $f(\mathbf{W}; x_i)$ is the neural network predicted value between [0, 1] for shot gather sample *i* (the 2D shot gather has been reshaped to a 1D vector in this formulation).

After the training stage, performance is measured by the mean Intersection over Union (mIoU) of the post-FB generated events class, with values from zero to one, where one is a perfect score. The Intersection over Union (IoU) measures the similarity between two sets. In the case of object detection or image segmentation, the Intersection over Union (IoU) score is defined by the fraction of the area of overlap A ∩ B over the area of union

$$IoU(A, B) = \frac{A \cap B}{A \cup B}$$

where A and B are two areas. The IoUreveals how similar the prediction is to the truth. It is a metric widely used for scoring and comparing image-segmentation method (Rahman and Wang, 2016). The mean Intersection-over-Union (mIoU) averages the IoUover the whole test set. The mIoU gives a more intuitive picture of segmentation quality.

The output, or prediction, of the trained neural network, is a per sample probability for the pixel to be of the post-FB class. Before we derive the FB picks from the predicted samples values, we need to go from a predicted mask with values ranging from zero to one, to a binary mask where the values are either zero or one. This is done simply based on a threshold value of 0.5.In order to compare the predicted first-break pick with the 'true' first-break pick, the predicted post-FB mask needs to be transformed in a FB line. The transformation of

the segmented gather into a first break line can be achieved with a search algorithm. The search algorithm starts at the most left column, picks the first break and sequentially looks for the [0,1] pattern in the next column starting with the rows closest to the last first-break pick. The difference between the predicted and 'true' first break is measured in time samples and the mean and standard deviation of this difference are calculated.

III. Data

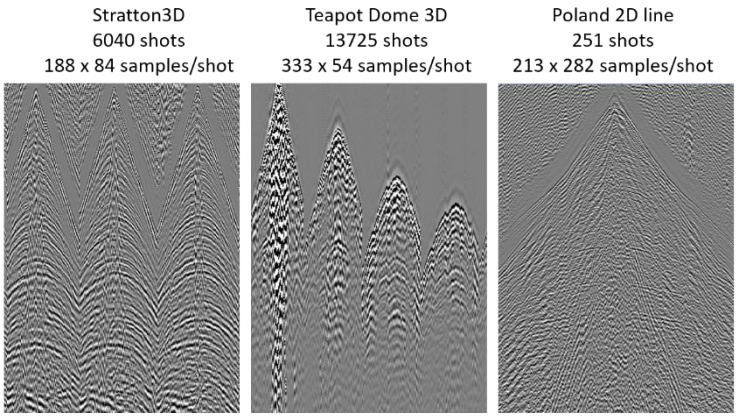

Figure no 9: Example gathers from the five input datasets. The samples per shot listed are time samples x geophone channels

For the training data we have used 20016 shot gathers from a mix of land seismic datasets: the Teapot Dome 3D (69%), Stratton3D (30%), Poland 2D seismic line (1%)from the SEG Wiki (https://wiki.seg.org/wiki/Open_data). Figure no 9 shows some shot gathers from these datasets. The masks for the segmentation problem are obtained by first running the algorithm by Yoo et al. (2019) and then using the obtained reference picks to generate binary masks. In order to increase the variety in the input data and thereby improve the neural network training, the shot gathers are perturbed or 'augmented' by randomly flipping positive and negative offsets and by randomly applying variations in seismic amplitude. In one test, we add random noise bursts to test robustness. The shot gathers have been truncatedto slightly more than the maximum time of the first-break in the data (1500-2500ms). Truncating the data serves two purposes. First, simply as data reduction. Secondly, it balances the number of samples in the two segments to about 50% per class. Prior to resampling to 8ms, we applied a 60Hz high-cut filter, which also serves to attenuate high-frequency noise. The resampling is done to decrease memory and time requirements for training and deploying the models. Preprocessing applied is a t^1 gain, amplitude clipping to the 95th percentile and amplitude standardization to zero mean and unit variance (i.e. a neural network friendly amplitude range). The data was resized to the nearest multiple of 64 samples larger than the original size in both directions and divided into patches of 64 x 64 samples. This solves the problem of varying shot gather dimension within and across dataset, since the models required a fixed input size. The input datasets are split randomly into 72% training, 8% validation and 20% testing subsets. All examples shown are from the test set. Training was done on a single NvidiaQuadro P5000 graphics card.

IV. Results

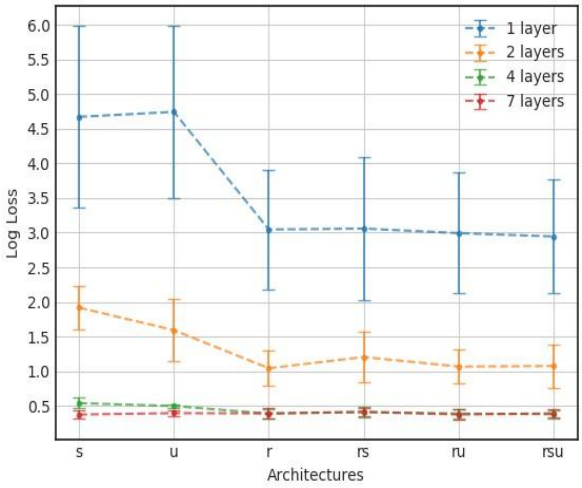

Figure no 10: The effect of neural network architecture and network depth as measured by the number of layers on the log loss. The standard deviation of tests per model is given. The effect of the architecture features on the log loss decreases with depth.

The effect of neural network depth

We tested what the effect is of a neural network depth of 1, 2, 4, and 7 layers for the SegNet, U-net, ResNet, ResNet+SegNet (RS), ResNet+U-net (RU), and Resnet+Segnet+U-net (RSU) architectures on all possible combinations of data sets. Figure no 11 shows the average log loss vs. number of layers. The effect of the number of hidden layers on the log loss is significant. The more hidden layers, the lower the log loss. The gain in improvement by adding more hidden layers decreases quickly. After increasing the network size to seven layers the improvements plateaus.

The effect of neural network architecture

In Figure no 10 we see that when networks are shallow, the effect of the type of network is more significant on the log loss values, in comparison to deeper networks. The presence of a ResNet architecture makes the biggest difference in shallow networks. The ResNet architecture actually contains additional layers, as was explained earlier: two convolution layers with a skip connection are considered as a single block, and only count as one layer. Not all architectures tend to make the same mistakes. Some tend to make more mistakes above the first break while other tend to make mistakes below the first breaks. Based on these observations we conclude that the seven layer U-net with ResNet block or skip connections give the best results. The remainders of the examples was made with this neural network architecture.

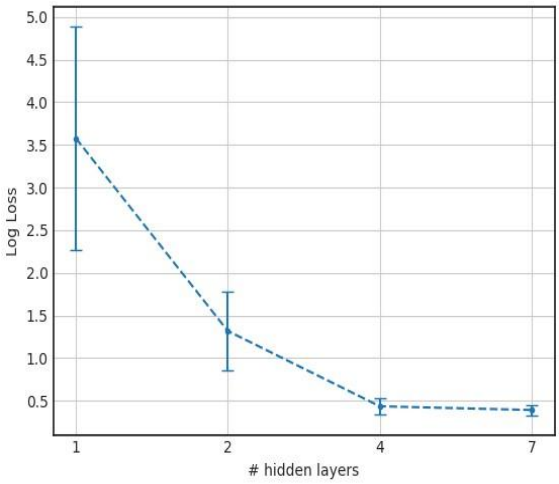

Figure no 11: The average loss for all the models, for all the different combinations of data sets. The uncertainty of the result is given by standard deviation of all test results per number (#) of hidden layers. This demonstrates that deeper layers perform better and more consistently.

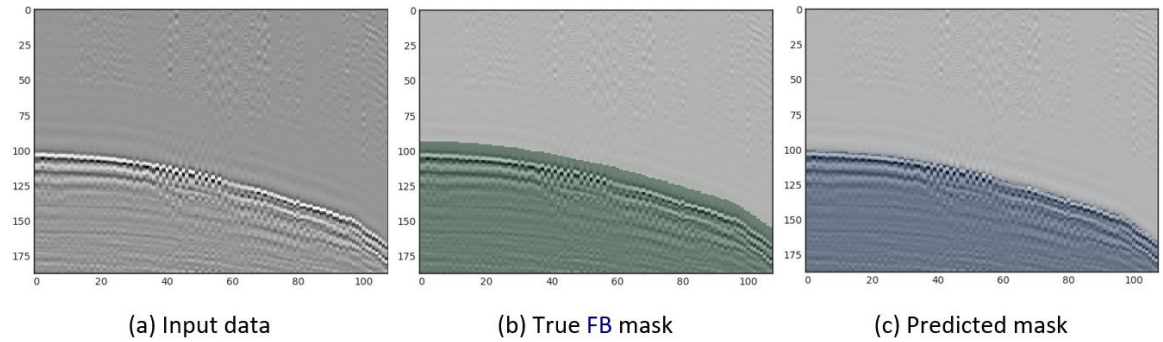

Figure no 12: Stratton data example with a) input gather, b) reference FB mask, c) predicted mask by the seven hidden-layer deep ResNet, SegNet, and U-net (RSU) architecture. Notice how the neural network prediction outperforms the conventional picks, illustrating the ability to handle outliers in the training data.

Figure no 12 shows the result on a shot gather from the Stratton dataset. The reference picks are by themselves an estimate and not true first-break picks, and here the true FB mask show a bias; it has picked the FB slightly too early. Figure no 12 shows that the CNN algorithm can improve such mispicks.

Ringing artifacts, AGC-boosted noise and weak amplitudes due to the spherical spreading of the energy, can be identified as sources for errors or misclassifications. We identify three regions of misclassifications: misclassifying pre-FB as post-FB, misclassifying post-FB as pre-FB, and making misclassifications near the FB. The latter was the least frequently occurring error and was mainly caused by problems due to splitting gathers into patches, but more on this later. Most errors made by the models are not near the first break. When misclassifications occur in the post-FB segment, they are often not near the first break. Sometimes, a complete patch is misclassified as can be seen inFigure no 13. Although some parts are not correctly segmented, it does not hinder the extraction of the first break (blue line) if one used a search algorithm starting at the proper First Break. This would be more problematic if wrongly segmented areas would interfere with the timing of the first break.

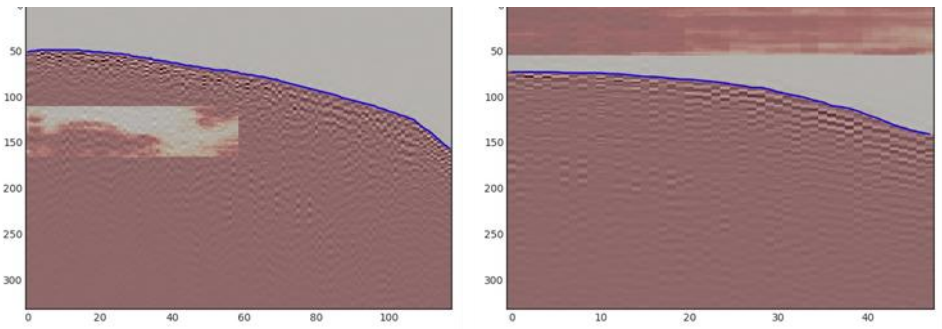

Figure no 13: The effect of segmentation on first break picks. Even when large errors are made during segmentation, it does not always translate in large errors in the first-break picks. These two figures show shot gathers from the Teapot test dataset with the predictions from a seven-layer-deep model with a ResNet, SegNet, and U-net (RSU) architecture. The model is trained on the Poland and Teapot training data subsets and tested on the Teapot test dataset. The mean Intersection over Union (mIoU) of this model on the Teapot test data subset is 0.993.

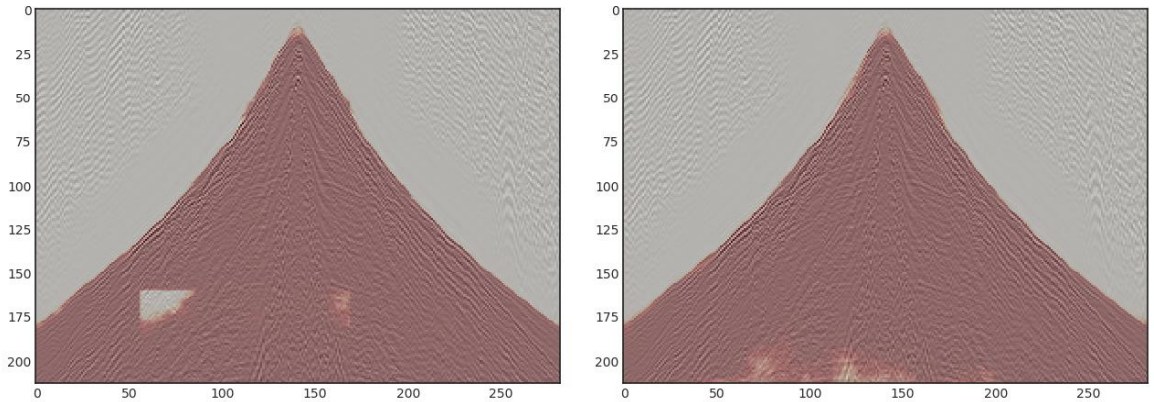

Figure no 14: The results of a seven layer deep RU model trained on 64 x 64 patches where the predictions where (left) applied on 64x64 patches and (right) the whole window.

The effect of dividing the gathers into patches of 64x64 pixels

We did the training and predictions on patches of 64x64 samples. However, this sometimes lead to misclassifications. Noise boosted by an Automatic Gain Control (AGC) can look like weak source-generated signal, and vice versa. For example, some patches of 64 x 64 pixels look like they only contain unorganized energy, such as in the ground-roll cone where high-energy scattered groundroll completely masks all coherent events. Such windows are likely to be characterized as the pre-FB class. In addition, the presence of ringing artifacts above the first break can look similar to weak multiples present in the data. In general, the models do not have a difficulty with these noise artifacts when enough context is given, or the presence of a strong event such as the first break is within the 64 x 64 pixels patch. When using a fully convolutional neural network (e.g. the U-net), models can be given any input size, which means we can train models on small patches and apply them to full sized shot gathers. See Figure no 14 for an example. The SegNet is not fully convolutional because the implementation of the unpooling layer forced a predefined input size.

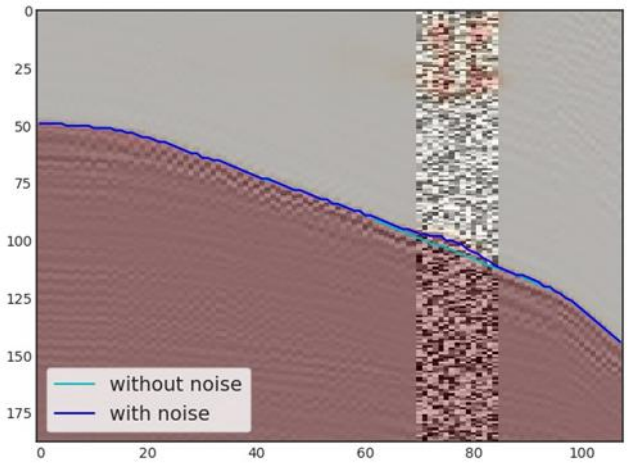

Figure no 15: Results of a seven layer RU architecture applied to unseen (i.e. not used in training) Stratton data set with and without the presence of a strong, 15 traces wide, noise burst.

Training with noise

In order to make the neural network algorithm more robust in FB picking on noisy data we have added strong (synthetic) noise on gathers after picking the FB with the reference algorithm(see**Figure no 15**). This is where most conventional 1D FB pickers fail and rely on post-processing techniques such as median filtering or more advanced approaches such as by Yoo et al. (2019). The results are very encouraging and suggest the algorithm can pick through a modest amount of noisy traces. The feature maps in **Figure no 6** and **Figure no 7** suggests how this works. The filters that are learned from the data have performed a decomposition into single and noise and were able to ignore effectively the noisy traces. In the upsampling (decoder) part of the network, the noisy areas are interpolated in a coarse sense. Presumably this works best with FB moveout that is piece-wise linear (hence the application in shot or receiver domain is preferred).

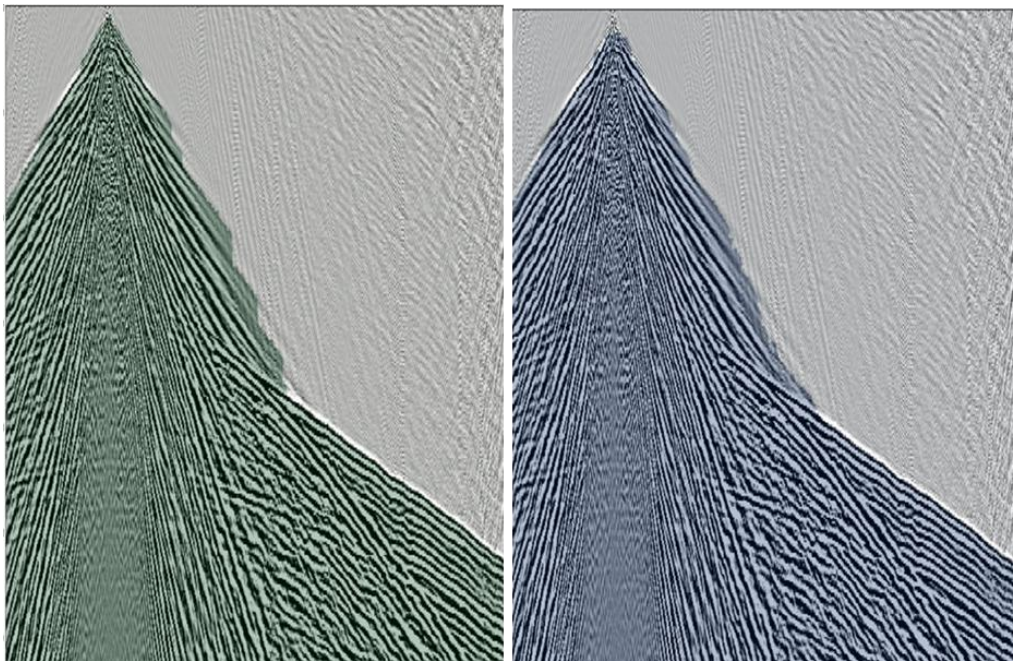

Figure no 16: Left: reference mask. Right: predicted mask. The three training dataset plus 100 shot gathers from this new dataset were used in training.

Application to unseen data

The test data that was used to compute the metrices was not used during training. However, it comes from the same dataset as the training and validation sets and therefore shares all the statistics and assumptions. Truly unseen data is data from which no shot gather were used during training and application to unseen data is therefore the only real test of generalization. Before applying the trained network to such unseen data,that unseen data must be preprocessed so that it looks as much as possible as the training data (in terms of amplitudes, noise levels, frequency content, gain, etc.). This is a hidden assumption of machine vision networks. Figure no 16 shows the results of applying a seven-layer RU network that was trained with all the previously mentioned data plus 100 shot gather from this new dataset. Application without these 100 gathers gave poor results due to the notably different structures in the noise prior to the first break and the presence of shingling (weak, fading refractions). However, by adding just 100 gathers to the training data the results improves significantly.

Figure no 17shows an example where no reference picks were available to re-train the network. In that case, pre-processing is crucial. In Figure no 17a, the OBN data set was only normalized and themodel behaved poorly, because of the large range of amplitudes, the first break appear much too weak compared to the training data. In Figure no 17b, the OBN data set was standardized by subtracting the mean and dividing by the standard deviation. This improved the performance of the model, but the presence of extreme amplitudes significantly affected the standard deviation and dampened everything except of the extreme amplitudes. A clipping of the 90th percentile, reducing the extremes on both ends before standardizing, proved useful. As can be seen in Figure no 17c, the seismic events below the first break are clearly visible. Adding a t^1amplitude gain correction prior to clipping(Figure no 17d) gave the best result. This shows that with some additional pre-processing steps a model that has been trained on Vibroseisland seismic can be applied on an OBN data set.

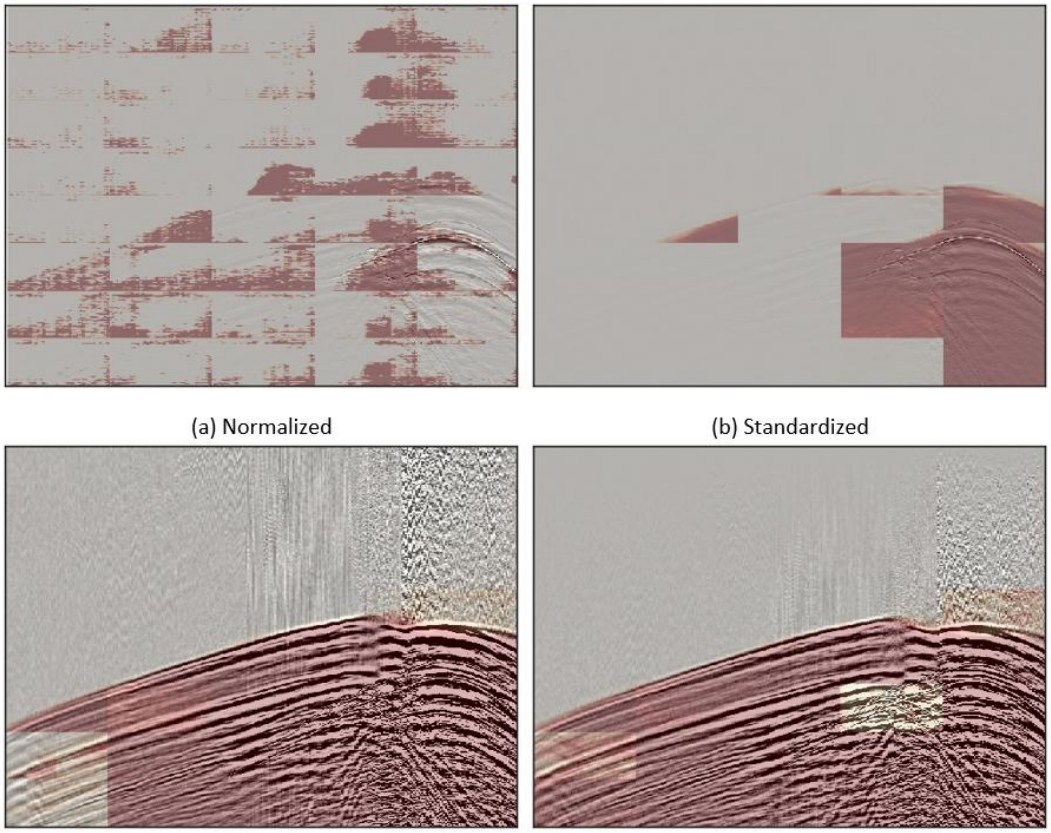

Figure no 17: The effect of pre-processing on FB picking on an Ocean-Bottom Node (OBN) gather. Especially extreme amplitude spikes, which are common in field data, need to be dealt with in order to make first-break picking work properly on a complete new and different data set without any additional training.

V. Discussion and Conclusion

We have shown that First Break picking can be handled as an image segmentation problem. The convolutional neural networks do a data dependent feature decomposition, which means they learn relevant patterns from the data. From the combination of these patterns, each sample gets assigned to the pre- or post-FB class. Because of this pattern-recognition capability,the trained network can generalize beyond the training data. From analysis of the filters that the convolutional neural network (CNN) has learned, a richness in observed patterns and signals can be seen. This suggests that CNNs are capable of more than segmenting samples in seismic data for first-break picking, and may also be able to classify a broader range of seismic events and noise, and even have applications in noise removal, as demonstrated by Baarman et al. (2018).

We have evaluated six different neural network architectures and obtained the best result with a seven layer U-net with residual blocks (a.k.a. skip connections). The effective receptive field of each sample in the final layer is determined by the number of convolutional and downsampling layers the model has. Increasing the effective receptive field is similar to increasing the field of vision of a human FB-picker. A downsampling layer has the most effect on the effective receptive field. The downsampling layers almost double the effective receptive field. Downsampling layers also perform a data decimation that results in a coarser output reducing the effect of noisy traces. One could interpret the coarser output as effectively zooming the picture out and looking to the larger structures. This is one of the explanation why deeper models perform better, can pick through noise, and are more robust. The robustness can also be concluded from the fact that the log loss and standard deviation decreases when more layers are added to the models.

The trained U-net model predicts the first break well even when noisy traces are added. The tests show that the trained models work also onunseen data, but only when thatdata has the same character and type of events as the training data. A side effect of the capability to pick through noisy traces is that the model will return a smoother first break pick when encountering static jitter.

An interesting finding is that the models can improve on the given training data when there is a small number of mispicks or random errors. This suggests machine vision based FB picking can improve picks from conventional methods. The robustness and generalization power of a model increases when it is trained with more data, although it does not need thousands of gathers to reach good results. A pre-trained model can be re-trained with a limited number of new gathers (in our example 100 shot gathers). This is similar to the learning process of a human first-break picker, who also needs a few examples to learn how to pick a new dataset.

Acknowledgements

We acknowledge the creators of the following Open Seismic Datasets: (1) the RMOTC and the U.S. Department of Energy for the Teapot Dome 3D seismic data and (2) the Bureau of Economic Geology at the University of Texas at Austin for the Stratton 3D survey.

References

[1.] Baardman, R., Classification And Suppression Of Blending Noise Using CNN, Conference Proceedings, First EAGE/PESGB workshop on Velocities, Feb 2018, https://doi.org/10.3997/2214-4609.201803017

[2.] Colombo, D., Federico M., Sandoval-Curiel, E., and Rovetta, D., 2016. "pQC: A Novel Approach for Robust Automatic near-Surface Analysis in Low-Relief Geology". The Leading Edge 35 (11): 952–60. https://doi.org/10.1190/tle35110952.1.

[3.] Duan, X., Zhang, J., Liu, Z., Liu, S., Chen, Z., and Li., W., 2018. "Integrating Seismic First-Break Picking Methods with a Machine Learning Approach". In SEG Technical Program Expanded Abstracts 2018. Society of Exploration Geophysicists. https://doi.org/10.1190/segam2018-2998293.1.

[4.] He, K., Zhang, X., Ren, S., and Sun, J., 2016. "Deep Residual Learning for Image Recognition". In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE. https://doi.org/10.1109/cvpr.2016.90.

[5.] Hu, L., Zheng, X., Duan, Y., Yan, X., Hu, Y., and Zhang, X., 2019. "First-Arrival Picking with a U-Net Convolutional Network". GEOPHYSICS 84 (6): U45–U57. https://doi.org/10.1190/geo2018-0688.1.

[6.] Kalashnikov, N. 2020. "Convolutional Neural Network for First Break Picking in Land Seismic Surveys". Conference Proceedings, 82nd EAGE Annual Conference & Exhibition Workshop Programme Volume 2020, p.1 - 5.

[7.] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

[8.] Krizhevsky, A., Sutskever,I., and Hinton, G.E., 2017. "ImageNet Classification with Deep Convolutional Neural Networks". Communications of the ACM 60 (6): 84–90. https://doi.org/10.1145/3065386.

[9.] Le, H. and Borji, A. (2017). What are the receptive, effective receptive, and projective fields of neurons in convolutional neural networks? CoRR, abs/1705.07049

[10.] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P., 1998. "Gradient-Based Learning Applied to Document Recognition". Proceedings of the IEEE 86 (11): 2278–2324. https://doi.org/10.1109/5.726791.

[11.] Long, J., Shelhamer, E., Darrell., T., Fully convolutional networks for semantic segmentation. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015.

[12.] Loginov, G., Anton, D., Litvichenko, D. and Alyamkin. S., 2019. "The First-Break Detection For Real Seismic Data With Use of Convolutional Neural Network". In 81st EAGE Conference and Exhibition 2019. European Association of Geoscientists & Engineers. https://doi.org/10.3997/2214-4609.201901614.

[13.] Ma, Y., Fei, T. and Y. Luo. 2019. "A New Insight Into Automatic First-Arrival Picking Based on Reinforcement Learning". In 81st EAGE Conference and Exhibition 2019. European Association of Geoscientists & Engineers. https://doi.org/10.3997/2214-4609.201901615.

[14.] Ma, Y., Cao, S., Rector, J.W., and Zhang, Z., 2020. "Automated Arrival Time Picking Using ASample-Level Network". GEOPHYSICS, June, 1–40. https://doi.org/10.1190/geo2019-0792.1.

[15.] Ronneberger, O., Fischer, P., and Brox, T., 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In Lecture Notes in Computer Science, 234–41. Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28.

[16.] Sabbione, J. I., and Velis, D., 2010. "Automatic First-Breaks Picking: New Strategies and Algorithms". GEOPHYSICS 75 (4): V67–V76. https://doi.org/10.1190/1.3463703.

[17.] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". The Journal of Machine Learning Research 15 (1): 1929–58. https://doi.org/10.5555/2627435.2670313.

[18.] Tsai, K. C., Hu, W., Wu, X., Chen J., and Han, Z., 2018. "First-Break Automatic Picking with Deep Semi-supervised Learning Neural Network". In SEG Technical Program Expanded Abstracts 2018. Society of Exploration Geophysicists. https://doi.org/10.1190/segam2018-2998106.1.

[19.] Xie, P., Boelle, J., and Blais. C., 2018. "Automatic Seismic First Arrival Picking With Deep-Learning". In First EAGE/PESGB Workshop Machine Learning. EAGE Publications BV. https://doi.org/10.3997/2214-4609.201803023.

[20.] Xie, T., Zhao, Y., Jiao, X., Sang, W., and Yuan, S., 2019. "First-Break Automatic Picking with Fully Convolutional Networks and Transfer Learning". In SEG Technical Program Expanded Abstracts 2019. Society of Exploration Geophysicists. https://doi.org/10.1190/segam2019-3215277.1.

[21.] Yoo, J., Borselen, R.V., Mubarak, M.S. andTsingas. C., 2019. "Automated First Break Picking Method Using a Random Sample Consensus (RANSAC)". In 81st EAGE Conference and Exhibition 2019. European Association of Geoscientists & Engineers. https://doi.org/10.3997/2214-4609.201901195.

[22.] Yuan, P., Hu, W., Wu, Chen, J., and Nguyen, H.V., 2019. "First Arrival Picking Using U-Net with Lovasz Loss and Nearest Point Picking Method". In SEG Technical Program Expanded Abstracts 2019. Society of Exploration Geophysicists. https://doi.org/10.1190/segam2019-3214404.1.

[23.] Rahman, M.A., and Wang, Y., 2016. "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation". In Advances in Visual Computing, 234–44. Springer International Publishing. https://doi.org/10.1007/978-3-319-50835-1_22.