Title

Detection of Intracranial Hypertension using Deep Learning.

Authors

Quachtran, Benjamin
Hamilton, Robert
Scalzo, Fabien

Publication Date

2016-12-01

DOI

10.1109/ICPR.2016.7900010

Peer reviewed

HHS Public Access

Author manuscript

Proc IAPR Int Conf Pattern Recogn. Author manuscript; available in PMC 2017 September 19.

Published in final edited form as:

Proc IAPR Int Conf Pattern Recogn. 2016 December ; 2016: 2491–2496. doi:10.1109/ICPR.2016.7900010.

Detection of Intracranial Hypertension using Deep Learning

Benjamin Quachtran[1], **Robert Hamilton**[2], and **Fabien Scalzo**[1]

[1]Department of Computer Science and Neurology, University of California, Los Angeles (UCLA)

[2]Neural Analytics, Inc. Los Angeles, CA

Abstract

Intracranial Hypertension, a disorder characterized by elevated pressure in the brain, is typically monitored in neurointensive care and diagnosed only after elevation has occurred. This reaction-based method of treatment leaves patients at higher risk of additional complications in case of misdetection. The detection of intracranial hypertension has been the subject of many recent studies in an attempt to accurately characterize the causes of hypertension, specifically examining waveform morphology. We investigate the use of Deep Learning, a hierarchical form of machine learning, to model the relationship between hypertension and waveform morphology, giving us the ability to accurately detect presence hypertension. Data from 60 patients, showing intracranial pressure levels over a half hour time span, was used to evaluate the model. We divided each patient's recording into average normalized beats over 30 sec segments, assigning each beat a label of high (i.e. greater than 15 mmHg) or low intracranial pressure. The model was tested to predict the presence of elevated intracranial pressure. The algorithm was found to be 92.05± 2.25% accurate in detecting intracranial hypertension on our dataset.

I. Introduction

In patients suffering from traumatic brain injuries, intracranial pressure is a metric closely monitored because, during hypertension episodes, they become susceptible to a host of other serious complications. Standard treatment of hypertension requires fluid drainage through lumbar puncture or surgically placed CSF valves, should drug therapy be unsuccessful or not recommended. Through our Deep Learning model, we hope to provide a framework for clinicians to treat hypertension in a proactive manner.

Recent improvements in computer hardware and interest in big data have led to advancements in machine learning. One sub-field of machine learning that holds immense promise for biomedical imaging applications is Deep Learning [1], [2]. It has proven to be an effective method of pattern recognition and has been applied to a wide variety of problems, including handwritten character recognition [3], face detection [4], anatomical classification [5] and speech recognition [6]. These systems could provide valuable inputs to physicians in terms of computer-aided diagnosis, image segmentation, image annotation, image registration, and multimodal image analysis. Convolutional Neural Networks (CNN), a form of Deep Learning, has been shown to excel in categorization of data through learning of characteristic features [7]. The use of convolution is especially advantageous for imaging and signal analysis because there is no reliance on specific spatial arrangements of inputs, allowing patterns to be recognized regardless of misalignment. Its application in the medical

field, however, is limited because of low training efficiency as well as implementation complexity. The advent of modular methods of Deep Learning has largely solved these issues, giving researchers the ability to utilize these algorithms while minimizing development time.

Numerous studies have been conducted suggesting use of machine learning methods and probabilistic frameworks to understand the characteristics in waveform morphology linked to elevated intracranial pressure [8], [9], [10]. A study by colleagues at UCLA demonstrated key relationships between ICP sub-peaks and hypertension [11]. Using the Morphological Clustering and Analysis of Intracranial Pressure (MOCAIP) algorithm, they were able to identify 24 metrics to be used as inputs to a quadratic classifier function. This method, however, requires manual search over a large number of features to find an optimal solution. Further studies suggested a more efficient use of MOCAIP through Linear Regression Analysis in conjunction with Randomized Decision Trees to generate a predictive model an AUC score of 0.98 [12]. Accessibility remains an issue because these models are not easily understood by bedside clinicians, limiting their practical use.

Detection of elevated ICP is a clinically relevant problem in that, for patients suffering from traumatic brain injuries, alarms are utilized to alert nurses of elevated pressure levels. These bedside alarms can have high false positive rates, taking valuable time away from clinicians and creating alarm fatigue [9]. An accurate method for detecting elevated ICP levels would enable clinicians to spend less time addressing false alarms, increasing overall response and treatment times in Intensive Care Units.

In this paper, we investigate the use of CNN's to automatically extract features in waveform morphology that are linked to intracranial hypertension. By not defining the feature set, we give the system the freedom to recognize any properties that are characteristic of hypertension, providing an unbiased analysis of the waveforms. A CNN operates by training on labeled data to create filters that, when convolved with new inputs, generate outputs that detect proper hypertension classification. Following training, these filters and their respective outputs can be further analyzed to understand what patterns the system is identifying, perhaps revealing previously unknown connections between morphology and hypertension.

II. Methods

A. Dataset Acquisition and Properties

The dataset originates from the University of California, Los Angeles (UCLA) Medical Center, with approval from the institutional review board (IRB) for use in this study. This is a retrospective study on patients who were being treated for various intracranial pressure related conditions including idiopathic intracranial hypertension, Chiari syndrome, and slit ventricle patients with clamped shunts. A total of 60 patients were considered for this study and their ICP and electrocardiogram (ECG) signals were recorded continuously. ICP was sampled continuously at 400 Hz using an Codman intraparenchymal microsensor (Codman and Schurtleff, Raynaud, MA) placed in the right frontal lobe. An expert researcher retrospectively identified intracranial hypertension (IH) episodes and annotated the time of

the elevation onset, elevation plateau, and invasive cerebrospinal fluid drainage in each patient recording. Within our cohort, 30 patients did not present any IH episodes and were excluded from the study. An additional 5 patients were excluded due to signal drop and artifacts that did not let the expert identify IH episodes with a high level of confidence. A total of 70 IH episodes were extracted from the ICP signal of the remaining 25 patients. Each segment included 20-minute of data, capturing the transition from a state of normal (0–15 mmHg) to elevated ICP (> 15 mmHg). The segments were time-aligned such that they contain 15 min of data before the plateau and 5 min after.

The data used for training and testing consisted of 89, 174 ICP beat samples collected from 25 patients. Figure 1 gives a visualization of a single ICP beat sample. Prior to normalization, the labels were generated based on each beat's pressure level; 1 for levels above 15 mmHg and 0 for those below, leading to a binary classification problem. Each beat sample was scaled to a length of 200 time points and normalized to range [0, 1].

B. Normalization

Because the ICP beat segments are of variable size, it is necessary to scale each signal $f(x)$ to a uniform length without losing any pertinent information. We make use of Cubic interpolation to accomplish this, providing a higher order precision than that of linear approximation [13]. The scaled output $f_{scaled}(x)$, defined over a new interval $[a, b]$ with N points, is calculated as a convolutional operation,

$$f_{\text{scaled}}(x) = \sum_{k-1}^{N+1} c_k * u\left(\frac{x - x_k}{h}\right)$$

where the x_k, c_k, and the cubic kernel $u(\upsilon)$ are defined as

$$x_k = \begin{cases} a & k=0 \\ x_{k-1} + \frac{b-a}{N} & 1 \le k \le N-1 \\ b & k=N \end{cases}$$

$$c_k = \begin{cases} 3f(x_0) - 3f(x_1) + f(x_2) & k=-1 \\ f(x_k) & 0 \le k \le N \\ 3f(x_n) - 3f(x_{n-1}) + f(x_{n-2}) & k=N+1 \end{cases}$$

$$u(\upsilon) = \begin{cases} \frac{3}{2}|\upsilon|^3 - \frac{5}{2}|\upsilon|^2 + 1 & 0<|\upsilon|<1 \\ -\frac{1}{2}|\upsilon|^3 + \frac{5}{2}|\upsilon|^2 - 4|\upsilon| + 2 & 1<|\upsilon|<2 \\ 0 & 2<|\upsilon| \end{cases}$$

The beat segments are then normalized to range [0, 1] using rescaling,

$$\vec{Z}_i = \frac{\vec{S}_i - \min(\vec{S}_i)}{\max(\vec{S}_i) - \min(\vec{S}_i)}$$

generating for each segment a vector of values $[\vec{Z}_1 \ldots \vec{Z}_M]$.

C. Deep Learning Model

The main contribution of this work is to provide a framework for the hypertension detection that consists of a two major components: an autoencoder and a convolutional neural network. Autoencoders are a form of unsupervised learning that utilize neural networks to generate encoded representations of data and, if used as a pre-training method, have been shown to improve the performance of deep networks [14], [15]. This form of pre-training consists of training the autoencoder on beat samples and then using the generated layers to initialize a neural network, allowing it to perform supervised learning on these encoded representations.

D. Neural Network Components

1) Convolutional Layer—In constructing a Neural Network, we are able to generate learnable filters that can identify features linked to an input's proper classification. Convolutional layers can consist of many filters and, when stacked together, can expand the number of detectable features used for learning. For each layer's filter, the convolutional operation between filter W and input x is calculated as

$$f(\tau) = \sum_{m=0}^{N} x_m \cdot W_{\tau - m}$$

The activation function is then applied to $f(\tau)$, providing a standardized measurement of the convolutional outputs. Use of a rectifier function

$$y(x) = \max\ (0, x)$$

has been shown to create networks that converge faster than those utilizing sigmoidal or hyperbolic functions [7].

2) Max-Pooling Layer—The ability of CNN's to extract features can be computationally intensive due to the large number of parameters present. To combat this, max-pooling can be used to select only key values in a local region, disregarding other non-critical elements and thereby reducing the number of network parameters. The input, with length N, is divided into $\frac{N}{L}$ sub-regions, each of length L, and from each sub-region a single maximum value is sampled. The result,

$$M(i) = \max\ ([x_{i \times L}, x_{(i \times L)+1} \ldots x_{(i \times L)+L-1}])$$

is a down-sampling that maintains the relative locations of the local maxima.

3) Dense Layer—Dense layers operate through element-wise multiplication of an input x_m and a filter matrix w_{mn} that is summed and passed through an activation function.

$$O_n = f\left(\sum_m x_n w_{mn}\right)$$

These dense outputs can generate meaningful probabilistic results through use of a softmax activation, shown to be an effective output stage for classification in CNN's because its ability to characterize multi-class regressions [7], [16]. The softmax activation gives a likelihood function

$$P(y_j = k|x_j) = \frac{\exp(x_j \theta_k^T)}{\sum_{k=1}^{M} \exp(x_j \theta_k^T)}$$

for input x_j and model parameter vectors θ_k. A maximum likelihood estimator is then used to predict the correct label

$$L_j = \arg \max_k P(y_j = k|x_j)$$

E. Back-Propagation

Proper weight calibration can be achieved using a feed-forward back-propagation algorithm that is able to fine-tune each filter's parameters through gradient descent of an error function E. For predicted labels L_j and ground truths y_j, the categorical cross-entropy error is defined as

$$E = -\sum_j y_j \log L_j$$

Upon each feed-forward iteration, consisting of the layers mentioned in prior sections, the gradient of the error with respect to the layer weights is calculated using the chain rule

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial u_j} \frac{a u_j}{\partial w_{ij}}$$

where u_j and y_j are the inputs and outputs of a given network layer. The weights w_i are adjusted in the direction of the negative gradient, thus decreasing the error function's value.

F. Autoencoder Structure

Autoencoders consist of two stages: an encoder and a decoder that consist of opposite layer structures. An encoder transforms the data into feature space using a combination of convolutional, max-pooling, and dense layers. The output is then fed into a decoder that

attempts to reconstruct the original signal through upscaling and convolution. Upscaling is performed to reverse the effects of max-pooling and can be implemented using nearest-neighbor interpolation. Autoencoder training is performed in an unsupervised fashion, where the objective is to minimize the squared error $E = (\vec{X} - \vec{Y})^2$ between the original data $\vec{X}$ and reconstructed data $\vec{Y}$. Feed-forward back-propagation is used to perform layer-wise training of the autoencoder.

III. Experiments

A. Training Process

Following normalization of the ICP beats, the autoencoder was trained over 20 epochs with a learning rate of $\sigma = 0.01$ to create encoded outputs of size 20. Figure 2 shows the autoencoder's layer architecture, consisting of a encoder and decoder stage. The final weights of the encoder were used as pre-training for the neural network, using the encoded inputs to perform learning.

Our CNN model was constructed using 3 layers: 2 convolutional layers, and a dense output layer, with the convolutional layers each consisting of 10 filters with size 5. A grid search was used to determine the combination of these layer parameters that minimized training loss, finding that a large number of filters had little effect on detection accuracy.

The CNN layer structure described is detailed in Table I and visualized in Figure 2. A learning rate of $\sigma = 0.1$ was used to train the network in a supervised fashion on the beats and their respective binary labels over 50 epochs. The network was then used as a model for hypertension detection by applying the trained layers on new beat samples to retrieve a binary output.

B. Results

The CNN model described was trained and tested on 76,137 beat samples using 3-fold cross-validation. The data was divided into three equal sets of 25,379 samples where we performed training on two sets and testing on the remaining set, repeating the process for each set. The model was found to correctly detect elevated levels of ICP with $92.05 \pm 2.25\%$ accuracy. In terms of performance, we can see in Table II that use of an autoencoder for pre-training enhances the CNN's detection accuracy. Table III summarizes the binary classification metrics of the network on these test samples. These results indicate that Convolutional Neural Networks are effective in characterizing ICP waveform morphology for detection of intracranial hypertension.

The autoencoder weights (Fig 3.) appear to resemble segments of ICP beat waveforms, indicating features, as opposed to noise, are being learned. In Figure 4, we see the shape of the network's loss function decays exponentially towards an asymptote, confirming that the model is learning at an appropriate rate [17] [18]. The curve approaching this asymptote after 30 epochs indicates an appropriate number of training cycles were performed. We also see that the CNN's validation accuracy increases as loss decreases (Fig. 5), indicating that the network is learning and its predictive power rises as a result.

IV. Discussion

The results of our experiments indicate that deep neural networks are able to accurately detect intracranial hypertension in patients using waveform morphology. Among Deep Learning methods, CNN's provide the unique ability to extract features from a signal in a translationally invariant manner, allowing us to objectively analyze these data. Because of the capability of Deep Learning to generate complex functions, over-fitting becomes an issue during the training phase, preventing the model to generalize well over new data. Using autoencoder pre-training, we are able to guide the network to perform learning on waveform characteristics, which we know to be predictive of hypertension [19].

A major difficulty in designing both the autoencoder and CNN frameworks was the selection of layer parameters. For each convolutional layer, an arbitrary number of filters and filter sizes could be used, giving us hundreds of possible combinations to choose from. The lack of theoretical methods for layer parameter selection meant manual search was required. A systematic search over a wide range of parameters might lead to further improvement in accuracy and will be considered as future work.

While these results are promising, fully evaluating the model's performance requires a more extensive test set. Testing on an independent set does not ensure the data is uncorrelated and further analysis is needed to understand how generalizable the network is.

Despite these concerns, use of Deep Learning has demonstrated to be a viable approach for intracranial hypertension detection. The natural extension to the issue addressed is the ability to perform predictions prior to onset, analyzing instead waveforms of low ICP levels to predict when hypertension is likely to next occur. Clinically, such a feat would be ground-breaking and enable patients to be treated in a proactive manner, reducing the need for surgical procedures.

Accurately measuring intracranial pressure is also an invasive process, requiring a hole to be drilled into the skull for placement of a monitoring device. Non-invasive techniques for measuring ICP have been researched, one study suggesting the use of cerebral blood flow velocity (CBFV) as a predictor of ICP pulse shape [20]. Advancements in prediction methods and analysis of CBFV-ICP relationships could be the key to providing clinicians with truly non-invasive procedures for predicting and treating intracranial hypertension.

V. Conclusion

Neural networks, and Convolutional Neural Networks (CNN) in particular, have shown to be effective in learning properties of ICP beat waveforms to detect the presence of intracranial hypertension. Methods to characterize hypertension in a non-invasive manner have been extensively researched, but still have not been realized. We anticipate the Deep Learning model described in this paper to be a stepping stone towards achieving this goal.

Acknowledgments

F. Scalzo is supported by a AHA grant #16BGIA27760152 and his research laboratory received generous hardware donations from Intel, Gigabyte, and Nvidia.

References

1. Stier, N., Vincent, N., Liebeskind, D., Scalzo, F. Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. IEEE; 2015. Deep learning of tissue fate features in acute ischemic stroke; p. 1316-1321.
2. Vincent, N., Stier, N., Yu, S., Liebeskind, DS., Wang, DJ., Scalzo, F. Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on. IEEE; 2015. Detection of hyperperfusion on arterial spin labeling using deep learning; p. 1322-1327.
3. LeCun, Y., Jackel, LD., Bottou, L., Brunot, A., Cortes, C., Denker, JS., Drucker, H., Guyon, I., Muller, UA., Sackinger, E., Simard, P., Vapnik, V. Comparison of learning algorithms for handwritten digit recognition. In: Fogelman, F., Gallinari, P., editors. ICANN. Paris: EC2 & Cie; 1995. p. 53-60.
4. Osadchy M, LeCun Y, Miller M. Synergistic face detection and pose estimation with energy-based models. JMLR. 2007 May.8:1197–1215.
5. Roth HR, Lee CT, Shin H, Seff A, Kim L, Yao J, Lu L, Summers RM. Anatomy-specific classification of medical images using deep convolutional nets. CoRR. 2015 abs/1504.04003.
6. LeCun, Y., Bengio, Y. Convolutional networks for images, speech, and time-series. In: Arbib, MA., editor. The Handbook of Brain Theory and Neural Networks. MIT Press; 1995.
7. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012:1097–1105.
8. Scalzo F, Hu X. Semi-supervised detection of intracranial pressure alarms using waveform dynamics. Physiological measurement. 2013; 34(4):465. [PubMed: 23524637]
9. Scalzo F, Liebeskind D, Hu X. Reducing false intracranial pressure alarms using morphological waveform features. IEEE Transactions on Biomedical Engineering. 2013; 60(1):235–239. [PubMed: 22851230]
10. Scalzo F, Asgari S, Kim S, Bergsneider M, Hu X. Bayesian tracking of intracranial pressure signal morphology. Artificial intelligence in medicine. 2012; 54(2):115–123. [PubMed: 21968205]
11. Hu X, Xu P, Asgari S, Vespa P, Bergsneider M. Forecasting ICP elevation based on prescient changes of intracranial pressure waveform morphology. IEEE Trans Biomed Eng. 2010 May; 57(5):1070–1078. [PubMed: 20659820]
12. Scalzo F, Hamilton R, Asgari S, Kim S, Hu X. Intracranial hypertension prediction using extremely randomized decision trees. Med Eng Phys. 2012 Oct; 34(8):1058–1065. [PubMed: 22401795]
13. Keys RG. Cubic convolution interpolation for digital image processing. Acoustics, Speech and Signal Processing, IEEE Transactions on. 1981; 29(6):1153–1160.
14. Erhan, D., Manzagol, P-A., Bengio, Y., Bengio, S., Vincent, P. The difficulty of training deep architectures and the effect of unsupervised pre-training; International Conference on artificial intelligence and statistics; 2009. p. 153-160.
15. Erhan D, Bengio Y, Courville A, Manzagol P-A, Vincent P, Bengio S. Why does unsupervised pre-training help deep learning? The Journal of Machine Learning Research. 2010; 11:625–660.
16. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L. Large-scale video classification with convolutional neural networks; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014 Jun.
17. LeCun, YA., Bottou, L., Orr, GB., Müller, K-R. Neural networks: Tricks of the trade. Springer; 2012. Efficient backprop; p. 9-48.
18. Karpathy A. Convolutional neural networks for visual recognition. 2016 [Online]. Available: http://cs231n.github.io/neural-networks-3/.
19. Scalzo F, Hamilton R, Asgari S, Kim S, Hu X. Intracranial hypertension prediction using extremely randomized decision trees. Med Eng Phys. 2012 Oct; 34(8):1058–1065. [PubMed: 22401795]
20. Schmidt B, Klingelhöfer J, Schwarze JJ, Sander D, Wittich I. Noninvasive prediction of intracranial pressure curves using transcranial doppler ultrasonography and blood pressure curves. Stroke. 1997; 28(12):2465–2472. [PubMed: 9412634]

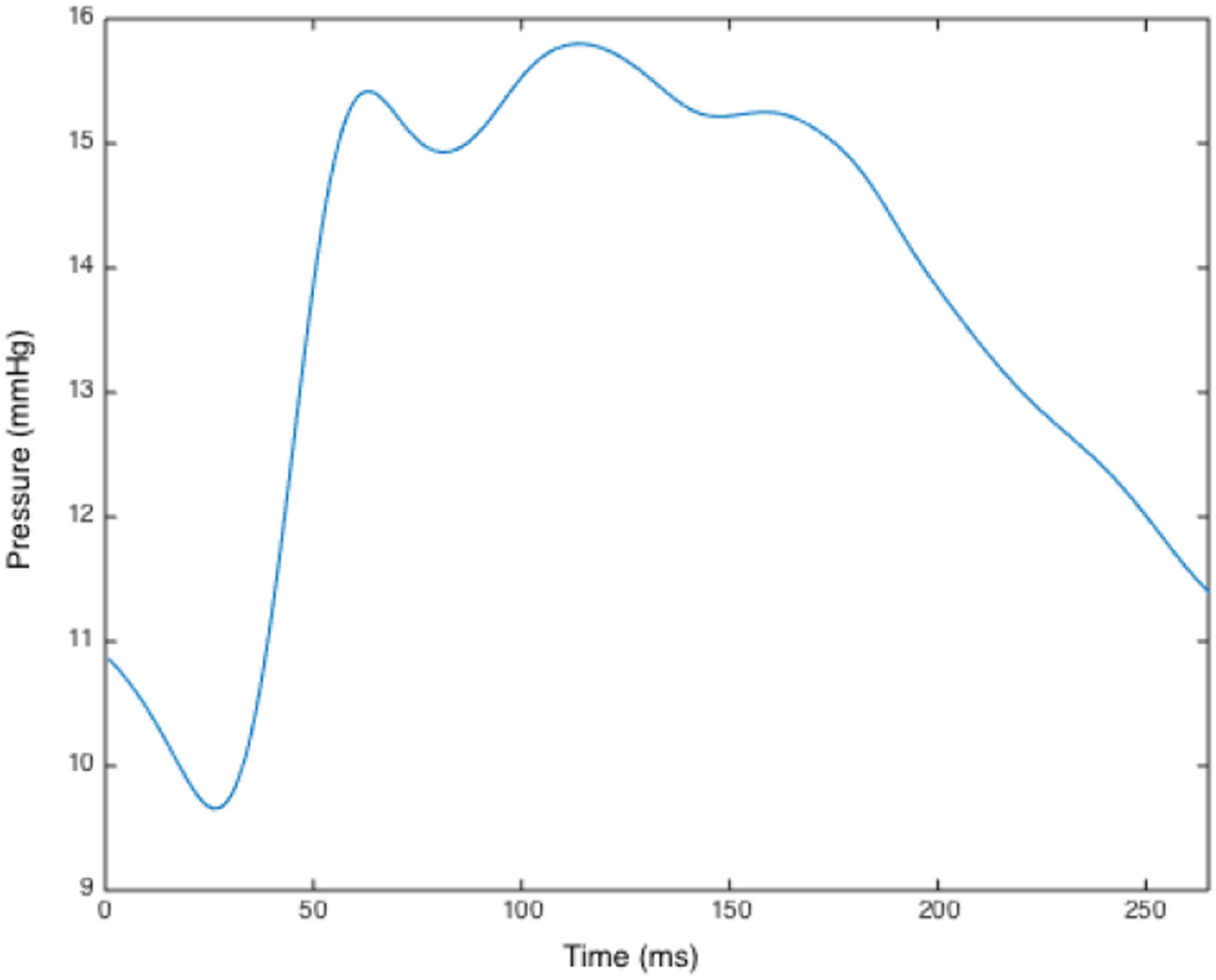

Fig. 1.
ICP waveform indicating pressure levels in mmHg versus time in ms.

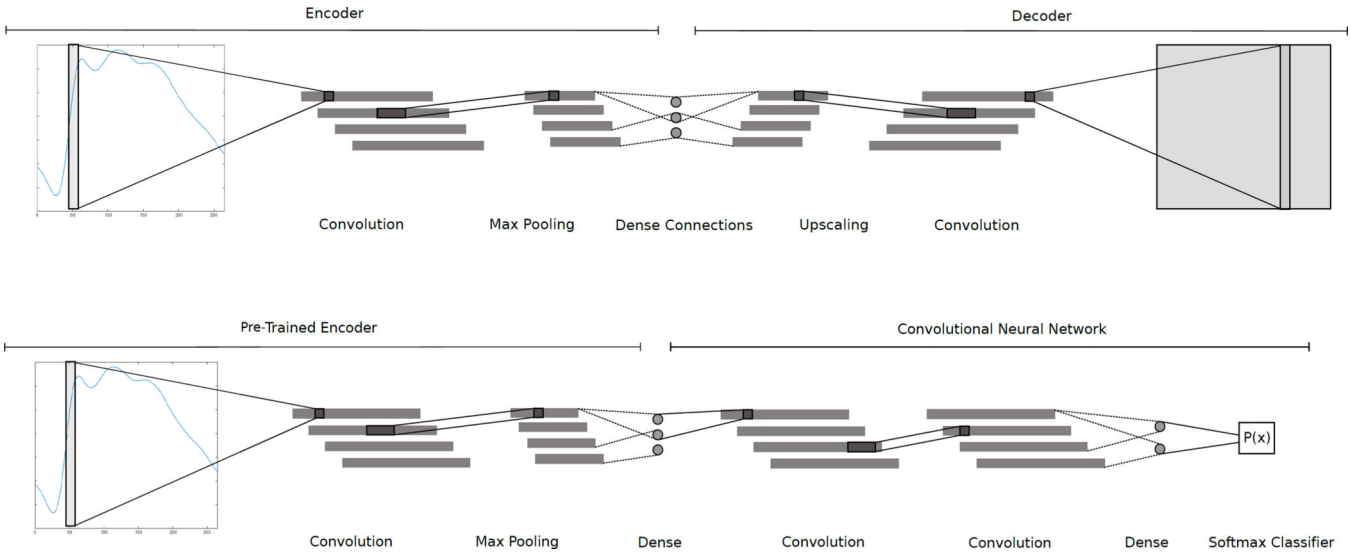

Fig. 2.

The top figure depicts the encoder and decoder stages used to train the autoencoder. The bottom figure is a full representation of the CNN detection model that utilizes autoencoder pre-training.

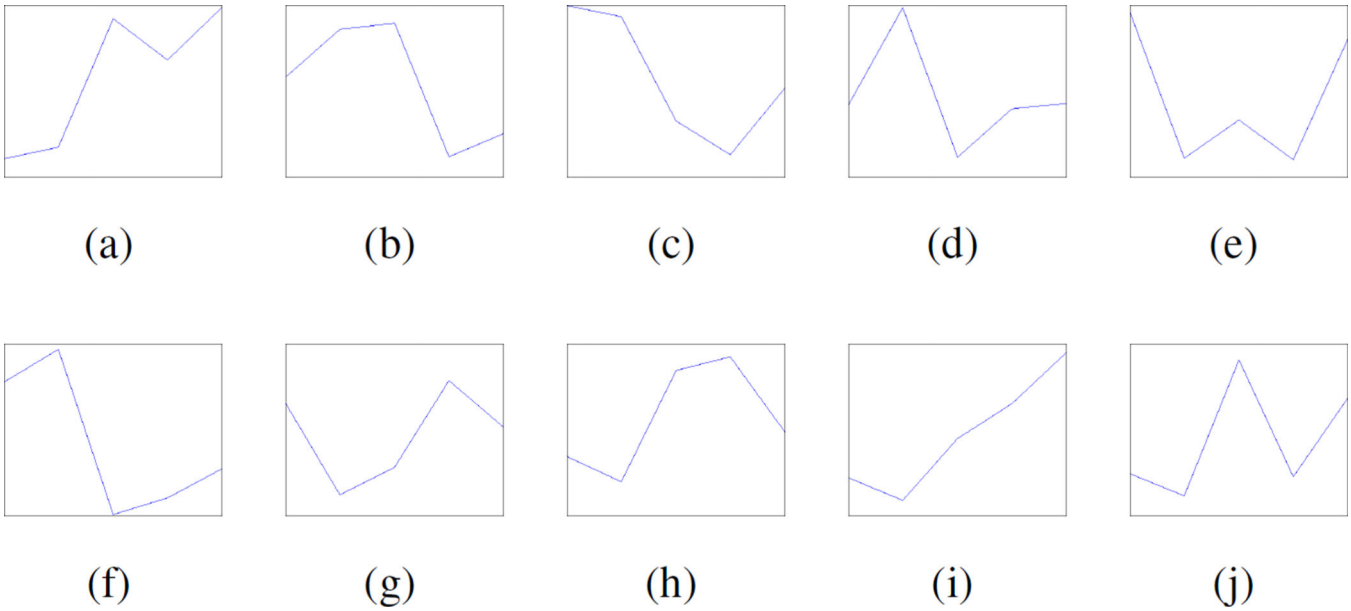

Fig. 3.
Encoder weights generated from training on ICP beat samples.

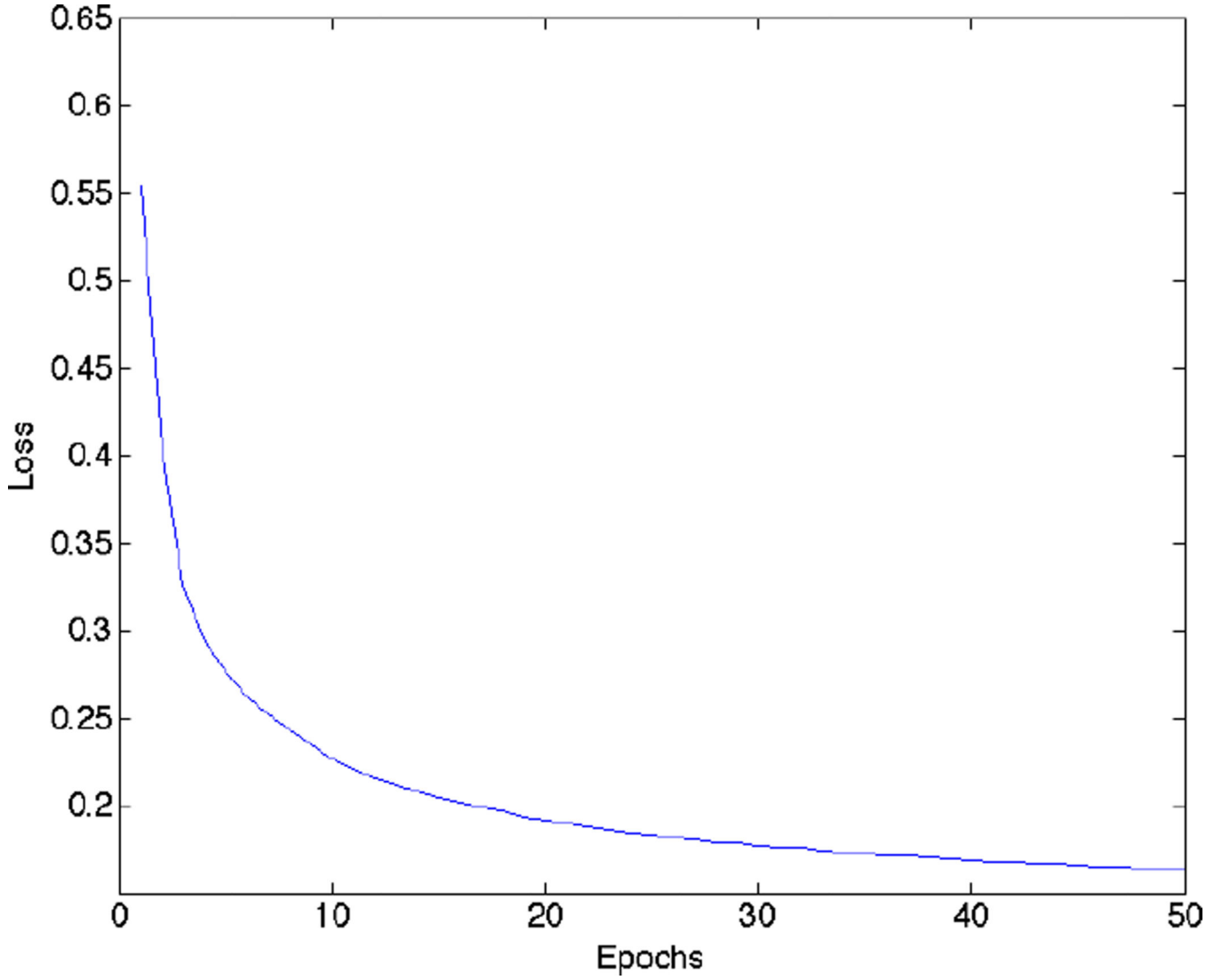

Fig. 4.

Training loss versus number of epochs completed. The curve begins to approach an asymptote after 30 epochs.

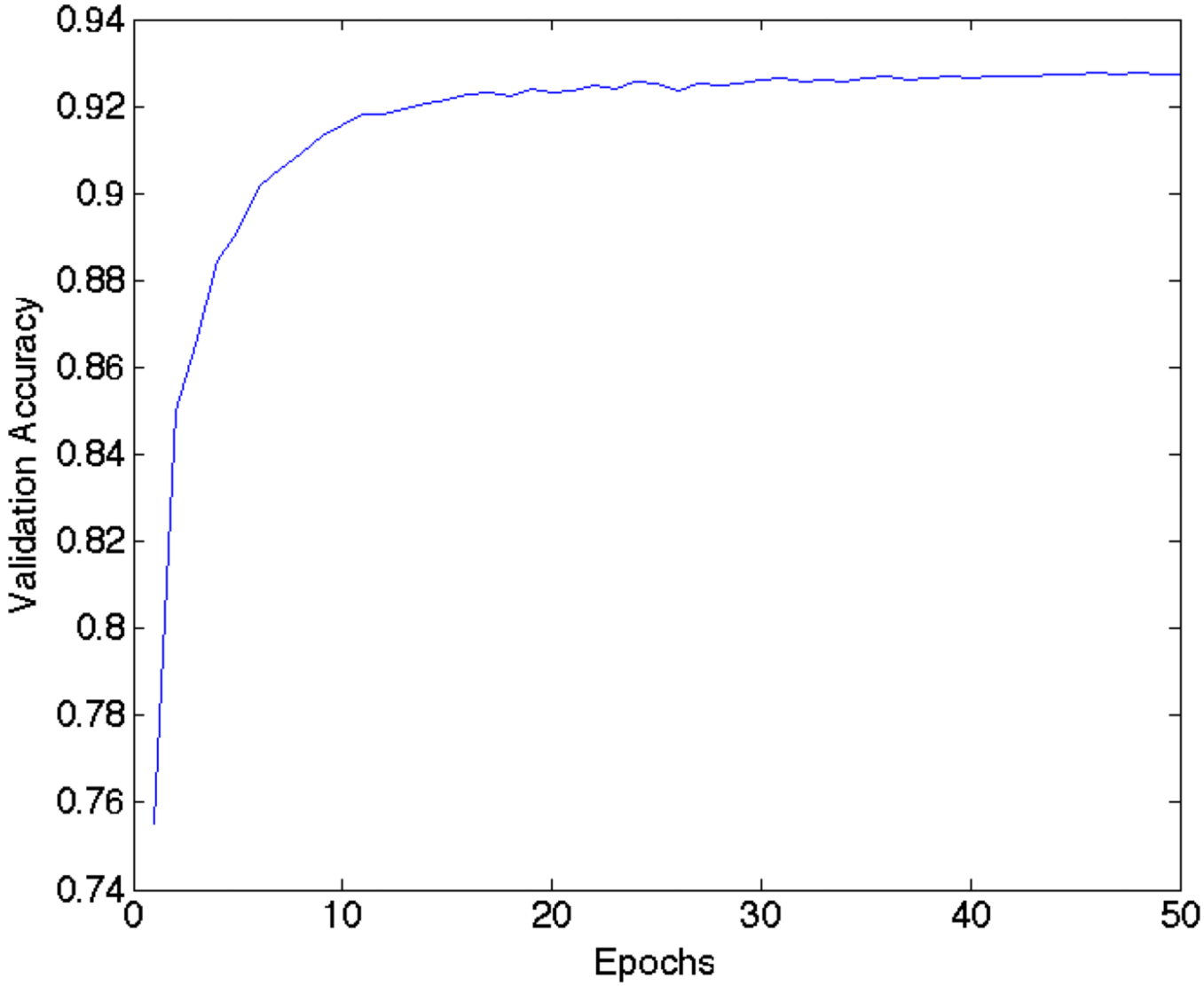

Fig. 5.
Validation accuracy versus number of epochs completed. The increasing curve indicates the model is learning as opposed to over-fitting.

TABLE I

Neural network layer architecture

Layer Type	Size
Input	200 samples
Convolutional	10 filters × 5 samples
Max-Pooling	4 samples
Dense	20 outputs
Convolutional	10 filters × 5 samples
Convolutional	10 filters × 5 samples
Dense	2 outputs

TABLE II

Comparisons of Model Performance

Learning Model	Accuracy
Autoencoder + CNN	**92.05%**
CNN	87.19%
Regression	73.55%

TABLE III

Results of the network on 7,694 beat samples

Metric	
True Positive Rate (TPR)	93.74%
True Negative Rate (TNR)	91.20%
False Positive Rate (FPR)	8.80%
False Negative Rate (FNR)	6.26%
Total Accuracy	92.05%