

Deep Learning Models for Automated Feature Extraction and Analysis

Tomasz Hachaj

Professor, AGH University of Kraków, Poland

Abstract

The increasing availability of large-scale digital datasets has created significant opportunities for advanced analytical systems while simultaneously increasing the complexity of extracting meaningful information from raw data. Conventional feature engineering approaches often require substantial domain expertise, manual intervention, and considerable computational effort, particularly when datasets contain high-dimensional, unstructured, or multimodal information. Deep learning provides an advanced alternative by automatically learning hierarchical representations and extracting relevant features directly from raw or minimally processed data. This study examines deep learning models for automated feature extraction and analysis and proposes an integrated conceptual framework for intelligent feature learning across complex data environments. The research adopts a conceptual and descriptive methodology based on secondary scholarly literature concerning deep neural networks, convolutional neural networks, recurrent neural networks, autoencoders, transformer architectures, representation learning, and automated analytical systems.

The proposed framework incorporates data acquisition, preprocessing, representation learning, automated feature extraction, feature optimization, analytical modeling, validation, and continuous learning. The analysis indicates that deep learning can reduce dependence on manual feature engineering while improving the ability to identify complex nonlinear representations in images, text, audio, time-series, and multimodal datasets. However, challenges involving data requirements, computational cost, interpretability, overfitting, model drift, and feature redundancy remain important. The study concludes that automated feature extraction through deep learning can strengthen analytical efficiency and scalability when supported by appropriate data preparation, model selection, validation, explainability, and continuous monitoring.

Keywords: Deep Learning, Automated Feature Extraction, Representation Learning, Neural Networks, Feature Engineering, Convolutional Neural Networks, Transformers, Intelligent Analytics

1. Introduction

The rapid growth of digital technologies has resulted in the generation of datasets characterized by increasing volume, dimensionality, diversity, and complexity. Modern applications generate information through sensors, cameras, enterprise platforms, mobile devices, scientific instruments, social networks, and connected systems. Extracting useful features from these datasets is a critical stage in analytical modeling because the quality of extracted representations directly influences downstream prediction and

classification performance. Conventional feature engineering typically depends on manually selected variables and domain-specific assumptions. Although these approaches can be effective for structured datasets, they become increasingly difficult to manage when information is high-dimensional, unstructured, or continuously generated.

Deep learning has introduced a major change in the way features can be extracted from complex data. Instead of requiring analysts to manually define all relevant characteristics, deep neural networks can learn representations through multiple computational layers. Lower layers can identify basic patterns, while deeper layers can combine these patterns into increasingly abstract representations. This hierarchical learning process allows deep learning systems to discover complex features directly from data. Consequently, deep learning has become an important methodology for image recognition, natural language processing, speech analysis, time-series prediction, anomaly detection, and other data-intensive applications.

Convolutional neural networks have demonstrated particular effectiveness in automated visual feature extraction. Through convolutional operations, these models can identify spatial patterns such as edges, shapes, textures, and more complex visual structures. Recurrent neural networks and related architectures can learn temporal dependencies within sequential information. Autoencoders can learn compressed representations while preserving important characteristics of the original data. More recently, transformer-based architectures have provided powerful mechanisms for learning contextual relationships within sequential and multimodal information. These architectures demonstrate the broad potential of deep learning for automated representation learning.

Automated feature extraction can provide several advantages over conventional feature engineering. It can reduce manual workload, improve scalability, and allow models to discover representations that may not be anticipated by human analysts. Deep learning can also combine multiple levels of representation within a single architecture, allowing raw information to be transformed progressively into analytical features. This is particularly valuable in applications involving complex visual, textual, acoustic, and temporal information. However, automated feature learning does not eliminate the need for careful data preparation and domain understanding.

The effectiveness of deep learning models depends substantially on data quality, model architecture, optimization strategies, and computational resources. Large neural networks can contain millions or billions of parameters and may require substantial training data and processing power. Overfitting can occur when models learn characteristics specific to training datasets rather than general patterns. Feature representations may also be difficult to interpret, creating challenges when users need to understand which characteristics influenced analytical results. These issues make model validation and explainability important components of automated feature extraction systems.

2. Objectives of the Study

2.1 Primary Objective

The primary objective of this study is to examine deep learning models for automated feature extraction and analysis and develop an integrated conceptual framework that connects raw data processing, representation learning, automated feature extraction, feature optimization, analytical modeling, validation, and continuous improvement.

2.2 Evaluation of Deep Learning Architectures

The study evaluates major deep learning architectures, including convolutional neural networks, recurrent neural networks, autoencoders, and transformer-based models, with respect to their capabilities for automated feature extraction from different forms of complex data.

2.3 Development of an Automated Feature Extraction Framework

The research aims to develop a conceptual framework that transforms raw or minimally processed information into meaningful learned representations. The framework integrates preprocessing, deep representation learning, feature optimization, analysis, and validation.

2.4 Identification of Implementation Challenges

The study identifies important challenges associated with automated feature extraction, including computational requirements, data dependency, overfitting, feature redundancy, interpretability, model drift, and deployment complexity. Appropriate mitigation strategies are also considered.

3. Review of Literature

3.1 Evolution of Feature Engineering

Feature engineering has traditionally represented an important stage of machine learning because algorithms frequently require meaningful input variables to generate effective predictions. Conventional approaches depend on domain expertise to identify, transform, combine, and select relevant characteristics. While manual feature engineering can produce useful representations, it can also be time-consuming and difficult to scale.

Deep learning introduced representation learning as an alternative in which models automatically learn useful characteristics from raw data. This transition has reduced dependence on manually designed features in many complex analytical applications.

3.2 Representation Learning

Representation learning focuses on automatically discovering transformations that make data more useful for subsequent analytical tasks. Deep neural networks can construct multiple layers of representation, with each layer capturing different levels of abstraction. This allows models to learn useful characteristics without requiring every feature to be explicitly defined by analysts.

Representation learning has become particularly important for images, language, speech, and other unstructured information. However, learned representations should be evaluated according to their ability to generalize beyond training examples.

3.3 Convolutional Neural Networks

Convolutional neural networks are widely associated with automated visual feature extraction because they can learn spatial relationships within images. Initial convolutional layers commonly learn relatively simple visual structures, while deeper layers capture increasingly complex patterns. Pooling and hierarchical processing can help reduce spatial complexity while retaining important information.

CNN-based feature extraction has been applied to image classification, object detection, medical imaging, quality inspection, and facial analysis. Their effectiveness depends on suitable architecture design, training data, and computational resources.

3.4 Recurrent Neural Networks

Recurrent neural networks were developed to process sequential information by maintaining information about previous observations. This makes them suitable for time-series data, speech, language, and other sequential applications. Long short-term memory and gated recurrent unit architectures were introduced to improve the ability of recurrent models to capture longer-term dependencies.

RNN-based systems can automatically extract temporal features from sequential observations. However, transformer architectures have increasingly become important alternatives for many sequence-processing tasks because of their ability to model long-range contextual relationships efficiently.

3.5 Autoencoders

Autoencoders are neural architectures designed to learn compact representations by reconstructing input data through an encoding and decoding process. The encoder transforms the original information into a lower-dimensional representation, while the decoder attempts to reconstruct the original input. The learned latent representation can capture important characteristics of the data.

Autoencoders have been applied to dimensionality reduction, anomaly detection, denoising, and unsupervised representation learning. However, the quality of the learned representation depends on architecture design and training objectives.

4. Research Methodology

4.1 Research Design

This study adopts a conceptual and descriptive research design to examine deep learning models for automated feature extraction and analysis. The research synthesizes established academic literature concerning deep neural networks, representation learning, convolutional neural networks, recurrent architectures, autoencoders, and transformers. The conceptual approach allows different deep learning architectures to be examined collectively rather than focusing on a single dataset.

The study emphasizes the relationship between automated feature extraction and downstream analytical performance. The resulting framework is intended to support future empirical validation.

4.2 Data Sources

The research uses secondary scholarly literature as its primary source of information. Relevant academic publications addressing deep learning, feature engineering, representation learning, neural network architectures, automated analytics, and intelligent data processing are considered. Literature from different application domains is examined to identify common principles associated with feature extraction.

The reviewed evidence includes methodological developments, application findings, advantages, and limitations. The information is organized thematically to support framework development.

4.3 Analytical Approach

A thematic and descriptive approach is used to analyze the selected literature. Studies are evaluated according to feature extraction capability, representation quality, analytical performance, computational requirements, scalability, interpretability, and adaptability. Different deep learning architectures are compared according to the type of data they are most suitable for processing.

Common implementation challenges are also identified. The findings are subsequently synthesized into an integrated automated feature extraction framework.

4.4 Framework Development

The proposed framework is structured around a sequence of interconnected analytical stages. Raw data are first acquired and preprocessed to improve consistency and reduce noise. The processed information is then supplied to an appropriate deep learning architecture for automated representation learning.

Extracted features are optimized and evaluated before being transferred to downstream analytical models. Model outputs are validated using suitable performance measures and domain knowledge. Continuous monitoring allows feature representations and model performance to be reassessed when data conditions change.

5. Proposed Deep Learning Feature Extraction Framework

5.1 Data Acquisition

The first stage of the framework involves collecting raw information from appropriate digital and physical sources. Depending on the application, data may consist of images, text, audio, video, sensor measurements, time-series observations, or multimodal information. The objective is to capture sufficient information to represent the underlying analytical problem.

Data acquisition should consider sampling quality, completeness, temporal consistency, and source reliability. Inadequate or biased data can affect the quality of learned representations. Appropriate metadata should also be retained to support later analysis and validation.

5.2 Data Preprocessing

Deep learning models can benefit substantially from appropriately prepared input data. Preprocessing may include normalization, resizing, tokenization, noise reduction, missing-value treatment, augmentation, and temporal alignment depending on the data type. The objective is to reduce irrelevant variation while preserving meaningful information. Data augmentation can increase diversity in training examples for some applications.

Table 1: Data Preprocessing Techniques for Deep Learning Models

Preprocessing Technique	Purpose	Contribution to Model Performance
Normalization	Standardizes numerical values into an appropriate range.	Improves training stability and convergence
Resizing	Adjusts image or input dimensions to a consistent format.	Ensures compatibility with model architecture

Tokenization	Converts textual data into meaningful computational units.	Facilitates effective language-data processing
Noise Reduction	Removes irrelevant or unwanted variations from input data.	Improves signal quality and predictive reliability
Missing-Value Treatment	Handles incomplete or unavailable observations.	Reduces data inconsistencies and information loss
Data Augmentation	Generates varied training examples from existing data.	Increases data diversity and model robustness
Temporal Alignment	Synchronizes observations collected at different time points.	Improves consistency in sequential analysis
Consistent Deployment Processing	Applies the same preprocessing procedures during training and deployment.	Supports reliable and stable model performance
Information Preservation	Prevents excessive transformation of meaningful characteristics.	Maintains relevant patterns and improves interpretability

Interpretation: Table 1 demonstrates that effective data preprocessing improves deep learning performance by standardizing inputs, reducing noise, addressing missing values, and increasing data diversity. Consistent preprocessing preserves meaningful information, strengthens model robustness, and supports reliable performance across training and deployment environments.

5.3 Automated Representation Learning

The representation-learning stage applies deep neural architectures to automatically identify useful characteristics within the processed data. CNNs can learn spatial features, recurrent architectures can capture temporal patterns, and transformers can learn contextual relationships. Autoencoders can generate compressed latent representations for unsupervised applications.

The choice of architecture should correspond to the structure and complexity of the underlying data. Effective representation learning reduces dependence on manual feature engineering while allowing models to discover higher-order relationships.

5.4 Feature Extraction and Optimization

Once representations are learned, relevant features can be extracted for downstream analytical tasks. Feature optimization may involve dimensionality reduction, feature selection, regularization, or latent-space analysis. The objective is to retain informative characteristics while minimizing redundancy and unnecessary computational complexity.

Automated feature extraction can produce representations that are difficult to describe using conventional variables but highly useful for predictive models. Validation is necessary to ensure that extracted features generalize beyond training data. Feature quality should ultimately be evaluated through downstream analytical performance.

6. Analysis

6.1 Automated Feature Discovery

Deep learning can discover features that may be difficult to define manually. Through hierarchical processing, models can transform simple input characteristics into increasingly complex representations. This is particularly useful for images and other high-dimensional data where relationships between individual variables may be complicated.

Automated discovery can reduce the time required for manual feature engineering and support scalable analytical workflows. However, the resulting representations may not always have intuitive interpretations. Performance evaluation is therefore necessary to determine whether discovered features provide meaningful analytical value.

6.2 Dimensionality Reduction

High-dimensional datasets can contain large numbers of variables, many of which may be redundant or weakly informative. Deep learning approaches such as autoencoders can transform these datasets into lower-dimensional latent spaces while retaining important information. Dimensionality reduction can decrease computational requirements and improve downstream modeling efficiency.

Latent representations may also reveal underlying structures that are difficult to identify from original variables. Nevertheless, reducing dimensionality can remove information if the representation is poorly designed. Validation should therefore determine whether important characteristics have been preserved.

6.3 Feature Quality and Predictive Performance

The usefulness of automatically extracted features can be assessed according to their contribution to downstream analytical performance. Strong representations should improve generalization, classification, prediction, clustering, or anomaly detection. Deep learning can identify complex nonlinear characteristics that may provide advantages over manually constructed features.

However, increased feature complexity does not automatically produce improved performance. Models can learn irrelevant patterns when training data contain noise or artifacts. Appropriate regularization, validation, and comparative testing are therefore essential.

7. Applications of Automated Feature Extraction

7.1 Image Analysis

Automated feature extraction is widely applicable to image analysis because visual data contain complex spatial patterns that are difficult to represent manually. CNNs can automatically learn features related to edges, shapes, textures, objects, and higher-level visual structures. These representations can support image classification, object recognition, defect detection, and medical image analysis.

Deep learning can also reduce dependence on manually designed visual descriptors. However, model performance may be affected by changes in image quality, lighting, orientation, and acquisition equipment.

7.2 Natural Language Processing

Deep learning can automatically extract semantic and contextual features from textual information. Transformer-based architectures can represent relationships between words, phrases, and larger sections of text. These representations can support classification, information extraction, sentiment analysis, document analysis, and question-answering applications. Automated language feature extraction reduces the need for manually constructed linguistic variables.

However, language is highly context-dependent and can contain ambiguity, domain-specific terminology, and implicit meanings. Careful evaluation is therefore necessary when applying learned representations to specialized domains.

7.3 Time-Series Analysis

Industrial sensors, financial systems, environmental monitoring platforms, and digital services generate continuous time-series information. Deep learning models can learn temporal patterns associated with trends, cycles, anomalies, and future conditions. Recurrent networks, temporal convolutional architectures, and transformers can extract representations from sequential observations.

These features can support forecasting, anomaly detection, predictive maintenance, and system monitoring. The effectiveness of temporal representations depends on sufficient historical information and stable relationships. Changes in operational conditions may require continuous model evaluation.

7.4 Medical and Scientific Data Analysis

Medical and scientific datasets frequently contain complex information that requires advanced feature extraction techniques. Deep learning can identify visual features in medical images, patterns in biological sequences, and relationships within high-dimensional scientific datasets.

Automated representation learning can accelerate analytical processes and reduce manual feature-design requirements. In scientific applications, learned representations can assist exploratory analysis and pattern identification. However, interpretability and validation are particularly important when extracted features contribute to high-impact scientific or clinical conclusions.

8. Discussion

8.1 Advantages Over Manual Feature Engineering

Deep learning can reduce the dependence on manually engineered features by allowing models to learn useful representations directly from data. This can save time and reduce the need for extensive domain-specific feature construction. Automated extraction can also identify nonlinear relationships that may not be captured through predefined variables. End-to-end architectures can integrate representation learning with prediction, reducing the need for separate analytical pipelines.

Nevertheless, domain knowledge remains valuable for defining objectives, validating outputs, and identifying inappropriate patterns. Deep learning should therefore complement rather than completely eliminate expert involvement.

8.2 Model Complexity and Interpretability

The complexity of deep neural networks can make extracted features difficult to interpret. A learned representation may contain useful information without having a simple human-readable meaning. This creates challenges when users need to understand why particular features contribute to a prediction. Visualization, attribution methods, attention analysis, and other explainability techniques can provide partial insights into model behavior.

However, explanations should be interpreted carefully because they may not fully represent internal computational processes. Transparent validation remains essential when deep learning models influence important decisions.

8.3 Computational Efficiency

Deep learning models can require substantial computational resources, particularly during training. Large architectures may involve millions of parameters and require specialized hardware. This can create challenges for organizations with limited infrastructure or applications requiring low-latency processing.

Model compression, pruning, quantization, efficient architectures, and edge computing can help address these limitations. Computational efficiency should be considered alongside feature quality and predictive performance. The most complex model is not always the most appropriate solution for a particular application.

8.4 Generalization and Robustness

A high-quality feature extraction model should generalize effectively to previously unseen data. Overfitting can occur when deep networks learn specific characteristics of training datasets rather than general patterns. Changes in data sources or environments can further reduce generalization.

Techniques such as regularization, data augmentation, transfer learning, and cross-validation can improve robustness. External validation using independent datasets can provide stronger evidence of generalization. Continuous monitoring is also necessary when the deployment environment changes over time.

9. Challenges and Limitations

9.1 Large Data Requirements

Deep learning models often require substantial quantities of data to learn reliable representations. Small or poorly labeled datasets can limit the ability of models to learn generalizable features. Data imbalance can further cause models to emphasize common patterns while underrepresenting less frequent conditions. Transfer learning and self-supervised learning can reduce some data requirements. Nevertheless, data quality remains more important than volume alone. Appropriate dataset design and validation are therefore essential for successful feature extraction.

9.2 Overfitting

Deep neural networks contain many parameters and can potentially memorize training examples. Overfitting occurs when learned representations perform well on training data but fail to generalize to

new observations. Regularization techniques, dropout, data augmentation, early stopping, and appropriate validation can reduce this risk. Model complexity should also be matched to dataset size and task requirements. Overfitting can be particularly problematic when automated features are subsequently used in high-impact analytical systems. Robust testing should therefore be incorporated throughout model development.

9.3 Feature Redundancy

Automated feature extraction can produce large numbers of representations, some of which may contain overlapping or redundant information. Redundant features can increase computational requirements and complicate downstream analysis. Dimensionality reduction and representation regularization can help control this problem. However, excessive feature reduction may remove information that contributes to prediction. Feature quality should therefore be evaluated based on analytical usefulness rather than quantity alone. Efficient representation learning remains an important research challenge.

9.4 Interpretability and Transparency

Deep learning representations are often difficult to describe using conventional human-readable variables. This creates challenges when users need to validate whether extracted features correspond to meaningful domain characteristics. Explainability techniques can provide information about influential inputs and internal representations, but their interpretation may remain complex. In high-impact applications, insufficient transparency can reduce user confidence. Research should therefore continue developing methods for interpretable representation learning. Transparent validation should accompany automated feature extraction throughout the system lifecycle.

10. Future Research

Future research should investigate more efficient deep learning architectures capable of performing high-quality feature extraction with reduced computational requirements. Lightweight neural networks, efficient transformers, model compression, edge-based inference, and energy-aware learning may support deployment in resource-constrained environments. Research should also examine self-supervised and unsupervised representation learning to reduce dependence on manually labeled datasets. Multimodal architectures that combine textual, visual, numerical, acoustic, and temporal information represent another important direction. These approaches could create richer feature representations for complex real-world analytical tasks.

Another important research area involves trustworthy and adaptive feature extraction. Future studies should develop methods for identifying when learned representations become outdated because of changing data distributions. Explainable representation learning could help users understand the characteristics captured by deep models. Research should also address fairness, robustness, privacy, adversarial resilience, and secure deployment of learned representations. Human-in-the-loop feature validation may provide an effective mechanism for combining automated discovery with expert knowledge. These developments can support more reliable, interpretable, scalable, and responsible deep learning systems.

11. Conclusion

Deep learning has transformed feature extraction by enabling computational systems to automatically learn hierarchical and complex representations from raw or minimally processed information. CNNs, recurrent architectures, autoencoders, and transformer-based models provide different mechanisms for extracting useful characteristics from visual, textual, sequential, and multimodal data. The proposed framework integrates data acquisition, preprocessing, representation learning, feature extraction, optimization, analytical modeling, validation, and continuous monitoring into a unified process.

The study demonstrates that automated feature extraction can reduce dependence on manual feature engineering and improve the scalability of data analysis. Deep learning models can identify nonlinear relationships and hierarchical structures that may be difficult to capture through conventional techniques. However, data requirements, computational complexity, overfitting, redundancy, interpretability, and model drift remain significant challenges. Successful implementation therefore requires careful model selection, robust validation, appropriate data governance, and continuous performance monitoring.

Overall, deep learning-based feature extraction provides a strong foundation for intelligent analytical systems operating in increasingly complex data environments. Future development should focus not only on increasing representation capacity but also on improving computational efficiency, transparency, robustness, adaptability, and responsible use. Combining automated feature learning with human expertise and systematic validation can enable organizations and researchers to obtain more reliable and meaningful insights from complex datasets. Such developments can strengthen the role of deep learning as a core technology for next-generation intelligent data analysis.

References

1. Hinton, G. E. (2000). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8), 1771–1800.
2. Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 19* (pp. 153–160). MIT Press.
3. Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527>
4. Ranzato, M., Poultney, C., Chopra, S., & LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems 19* (pp. 1137–1144). MIT Press.
5. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 1096–1103). ACM. <https://doi.org/10.1145/1390156.1390294>
6. Lee, H., Ekanadham, C., & Ng, A. Y. (2008). Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20* (pp. 873–880). Curran Associates.
7. Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1), 1–127. <https://doi.org/10.1561/22000000006>
8. Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics* (pp. 315–323). JMLR.

9. LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010). Convolutional networks and applications in vision. In *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems* (pp. 253–256). IEEE. <https://doi.org/10.1109/ISCAS.2010.5537907>
10. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates.
11. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014* (pp. 818–833). Springer. https://doi.org/10.1007/978-3-319-10590-1_53
12. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
13. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–9). IEEE.
14. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). IEEE. <https://doi.org/10.1109/CVPR.2016.90>
15. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431–3440). IEEE. <https://doi.org/10.1109/CVPR.2015.7298965>
16. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
17. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
18. Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7, 87–93. <https://doi.org/10.1007/s13735-017-0141-z>
19. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018, Article 7068349. <https://doi.org/10.1155/2018/7068349>
20. Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: A review. *EURASIP Journal on Wireless Communications and Networking*, 2017, Article 211. <https://doi.org/10.1186/s13638-017-0993-1>