Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 10 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i10.49037

PAPER

Handpose Estimation Based Learning Academy for Improving Typing Efficiency by Using Mobile Technologies

Mathias Mattersberger, Josef Wachtler, Martin Ebner(✉)

Educational Technology, Graz University of Technology, Graz, Austria

mebner@gmx.at

ABSTRACT

This paper focuses on enhancing touch-typing skills through state-of-the-art deep learning technologies. It has been demonstrated that writing posture can be assessed using integrated keyboard detection and hand pose estimation in real-time browser environments. A unique dataset was created for training the object detection model to recognize individual keyboards. The object detection models were trained using various architectures and then assessed for both inference time and accuracy. In addition, a hand pose estimation was implemented to precisely recognize 21 knuckle points of the hand and compute their exact positions. In order to implement object detection and hand pose estimation, a stream transmission of the keyboard and hand scene is required. For this purpose, server-client communication via a QR code connection is implemented to transfer the stream between two mobile devices. Based on these deep learning technologies, a mobile web app was developed that offers 170 learning courses for touch-typing training. With the help of the Typing Learning Academy prototype, a study was conducted as part of this paper to evaluate the usability and utility of the learning app. This research demonstrates the potential for enhancing the development of writing skills through touch typing through the utilization of advanced deep learning technologies.

KEYWORDS

mobile technologies, touch typing, learning web app, hand pose estimation, object detection, keyboard recognition, transfer learning, convolutional neural networks (CNN)

1 INTRODUCTION

Deep learning is one of the most frequently used terms in the context of current technological progress. This involves the use of multilayered artificial neural networks to learn from large amounts of data.

As a result, there are many fields of application, such as hand pose estimation and object detection technologies, where large amounts of data can be analyzed for correlations and patterns.

Mattersberger, M., Wachtler, J., Ebner, M. (2024). Handpose Estimation Based Learning Academy for Improving Typing Efficiency by Using Mobile Technologies. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(10), pp. 59–70. https://doi.org/10.3991/ijim.v18i10.49037

Article submitted 2024-01-11. Revision uploaded 2024-03-24. Final acceptance 2024-03-29.

Another significant aspect of this work focuses on touch typing. The touch system was introduced to increase typing efficiency and ensure more effective completion of computer tasks. By using touch-typing, one can achieve a healthier body posture and finger posture. These benefits clearly illustrate the importance of correct finger posture in the context of digital typing applications. In this paper, a unique touch-typing learning technology is presented that enables real-time monitoring of correct finger positions on the keyboard.

To ensure accurate tracking of finger positions on a keyboard, a QR code connection is implemented. By scanning a randomly generated QR code, the mobile phone's camera automatically connects to the web application. The scene captured by the keyboard is then transferred to the web app using the mobile phone camera and processed there. This stream is utilized to identify individual keys on the keyboard through trained object detection. Visible hand and finger movements are also detected in the stream, and specific knuckle positions are calculated. Based on these deep learning technologies, a web application with 170 learning courses is provided to ensure optimal progress in touch-typing. To evaluate the implemented learning academy, various testers with different technical setups are invited to thoroughly test the platform. This research study aims to answer the question of whether a more efficient learning process can be identified. The potential impact of this work is the initial implementation of software for real-time detection of finger positions within a touch-typing learning web application. In addition, a first-time dataset for annotating keys on the keyboard is published. This fully annotated dataset can be used for future applications of key detection. The aim of this paper is to showcase the developed app and apply research to address and resolve the following research questions:

One of the main questions is: "Is it possible to utilize hand pose estimation technologies and object detection technologies to validate accurate finger placement (touch typing) within a browser environment in real-time?" Finally, the study analyzes whether the learning progress in acquiring the touch-typing system can be enhanced more efficiently through the use of a web application that provides real-time feedback on finger positioning on the keyboard.

2 RELATED WORK

2.1 Deep learning

Artificial neural networks with multiple layers are utilized for this purpose. Artificial neural networks are algorithms that consist of multiple artificial neural nodes and at least three layers. The functioning of the human brain is very similar to the operation of artificial neural networks. It is essentially a mathematical model of the logic of our brain. It also has nodes, which are referred to as artificial neurons. The different layers can communicate with each other via these nodes. In this way, the output of the previous layer serves as input for the following layer. The nodes of the different layers can be connected and weighted as needed. These weighted nodes and connections constitute the intelligence of this artificial neural network. The greater the number of layers, nodes, and connections, the more complex, computationally intensive, and intelligent an artificial neural network becomes. These networks learn from large amounts of data and can recognize complex patterns and structures. Deep learning algorithms autonomously add new layers to models, thereby developing independent models. These models are trained with large amounts of data and can also work with unstructured data. Artificial neural networks are utilized for classification and regression problems. Well-known areas of application for deep learning include speech and face detection as well as image recognition.

2.2 Image recognition using convolutional neural networks

Convolutional neural networks (CNNs) are a special type of artificial neural network used specifically for grid data to analyze features in images. Convolutional neural networks are particularly suitable for tasks in the field of image and video analysis. Special layers such as the convolutional layer, the pooling layer, and the fully connected layer distinguish convolutional neural networks from conventional artificial neural networks. Computers use pixel matrices to interpret images. These pixel matrices pass through the layers (see Figure 1) to classify objects based on probability values between 0 and 1. In principle, convolutional neural networks typically consist of a series of convolutional and pooling layers, followed by one or more fully connected layers.

Convolution layers [1] extract features from the input image. In this process, the input image is divided into small squares, and the relationship between the pixels is preserved by learning the image features. The extracted image features are stored in a feature map, which is the output of the convolution layer.

The pooling layer reduces the dimensions of the feature map even further. Dominant features are extracted within a specific set of neighborhoods. Down sampling reduces the dimensionality without losing much information [2]. The advantages include reduced computational effort, decreased sensitivity of the network to various features, and the ability to create deeper and more complex networks.

The fully connected layer is now responsible for classifying various images, thus performing the classification process in this system. The image is converted into a vector and serves as input for the neural network. Features are combined in various layers of this network to create a model.

The output layer at the end of the CNN is responsible for categorizing the input image. The number of neurons in the output layer determines the number of different categories into which the image can be classified.

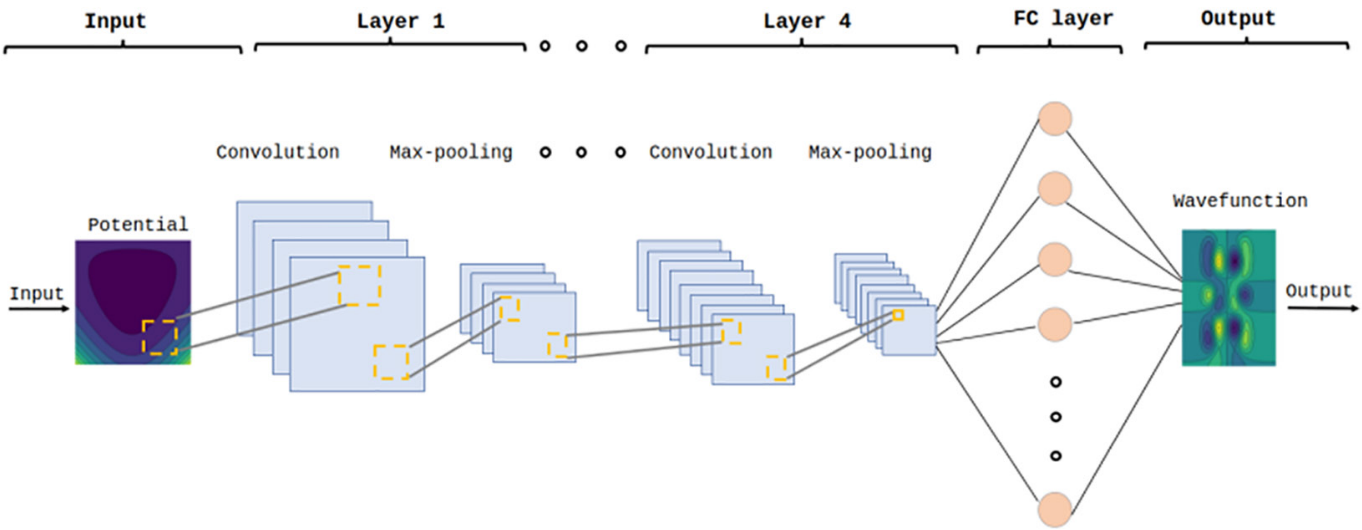

Fig. 1. Design of a convolutional neural network (CNN) [3]

2.3 Object detection

Object detection utilizes deep learning technologies and CNNs to identify objects in images and ascertain their positions. This involves classifying the objects present in the image and determining their locations. Object detection, therefore, consists of classification and localization [4]. Classification determines the class of the object in the input image. Several different classes can also be identified in the image. Localization determines the position of the object in the image. This is indicated in computer vision by so-called bounding boxes, which are drawn around the detected object. The input image is processed through the CNN to extract features such as edges or shapes. The output of this CNN process is then divided into a classifier and a regressor. The classifier categorizes the resulting vector into a known object class. The regressor determines the bounding box for the object that was classified earlier.

2.4 Hand pose estimation

Deep learning also enables completely new approaches to hand pose estimation technologies. In the past, more work was done with in-depth images. However, deep learning approaches have made it possible to estimate hand pose with a single RGB image using large datasets. Hand pose estimation involves estimating hand posture and hand parts by identifying specific points on each joint. The number of joints can vary depending on the trained model. Most models use 21 hand points (see Figure 2).

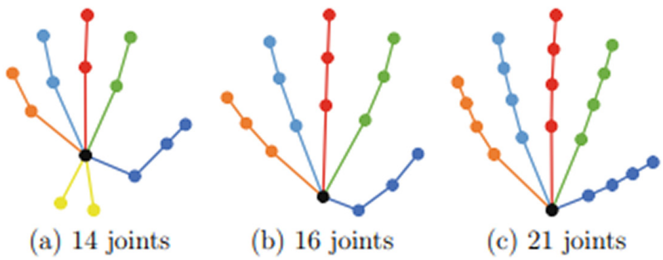

Fig. 2. Visualization of modeling hand with 14, 16, and 21 joints [5]

Very large datasets are essential for training networks with RGB images. In hand pose estimation approaches, the hand is first localized in the image, and then the image dimensions are reduced to the area containing the hand. This cropped area is then passed around the network to estimate the positions of the individual hand points and joints. The areas of application for hand pose estimation are very diverse. They are used for gesture control, sign language recognition, or to verify the correct execution of therapeutic exercises in medicine. As this paper shows, this technology can also be used in education for learning applications such as touch typing.

2.5 Improving touch typing with the help of learning apps

The most common method of typing words on a keyboard is the touch system. It was previously used for writing on typewriters and was first mentioned as an official writing method in 1988. The principle is to assign a specific key on the keyboard to each finger. Only this finger may be used for the corresponding key. The aim of touch typing is to make typing faster and more efficient. The natural hand position is enforced, leading to better ergonomics and reduced finger fatigue. Based on the results obtained in [6], it is shown that this learning software facilitates a more efficient learning of the touch system. The *Typing Learning Academy,* which utilizes deep learning methods, is the first learning software of its kind, and there is no related work in this area of research.

3 IMPLEMENTATION

3.1 Technical realization of the video stream transmission

The challenge is to implement a straightforward connection setup to enable video streaming of the keyboard scene for the user. The scene of the keyboard and hands will be recorded, transmitted, and further processed using the mobile phone's camera. To fulfill these requirements, a QR code stream transmission is used. The basic concept is that the QR code generated is scanned with a mobile phone, and the connection is established fully automatically without any further input or additional steps. After scanning the QR code, the stream is recorded using the mobile phone's camera and sent to the web app for detection and verification of the finger and key positions, followed by additional processing. Technologies such as SimplePeer for peer-to-peer stream transmission and Socket.io for establishing an initial connection between two clients are used to address the described issue. It is important to implement a straightforward and fully automated connection setup for the user. The QR code technology is ideally suited for this, as the entire connection setup process is automated, except for scanning the QR code.

3.2 Technical realization of object detection technology to detect keys on the keyboard

Object detection is a technology that utilizes machine learning and image processing methods to identify objects and instances. It pursues two main objectives. It classifies and localizes objects in an image. The object detection algorithm draws bounding boxes around the recognized objects. These objects are defined by labels and assigned to a class. The bounding box contains information about the position in the image where a detected object is located. The labels assigned to the binding boxes provide information about the objects they represent. To obtain this information, the image is split into small parts, called patches, and features are extracted from them.

These features contain important information from the image and are used for classification and localization. In summary, the image undergoes a process where the trained CNN extracts features and then conducts classification and localization.

Basically, two main categories can be distinguished between one-stage algorithms and two-stage algorithms. One-stage algorithms utilize a neural network model to predict objects simultaneously. You Only Look Once [7], Single Shot Multi Box Detector [8], and RetinaNet [9] are the best-known one-stage object detection algorithms. Two-stage algorithms consist of two steps. In the first stage, possible object areas are determined using a region proposal. In the second stage, these object areas are utilized in the neural network model for classification and localization. The best-known two-stage algorithms are R-CNN [10], Faster R-CNN [11], and Mask R-CNN [12].

Fig. 3. Annotated sample images of the data set for model training

To train these object detection algorithms, images are used as datasets. In these images, the positions of the objects are marked by a bounding box. In addition, the information about the object is defined by labels assigned to the bounding box. This image labeling is referred to as an annotation. In this work, a completely new dataset for recognizing keys on a keyboard is created as there is no existing research on this topic. Also, the object detection model in this paper, designed to detect keys on keyboards, has no related work and is the first of its kind. The generated dataset (see Figure 3), consisting of 5254 images, can be used for future work.[1]

In selecting the images (see Figure 3), care was taken to compile a comprehensive dataset that encompasses real-life scenarios and various application areas. To enhance the robustness and diversity of the dataset, data augmentation techniques were applied to the annotated images. This data augmentation technique involves using images from the dataset to generate new images. Data augmentation techniques, such as brightness modification and random cropping, were used.

Based on the dataset and the TensorFlow Object Detection API [13], a model was trained to detect keys on keyboards. TensorFlow offers an object detection API that provides various object detection models for training and refining one's own. The various models exhibit different mean average precision and inference times. Architectures such as CenterNet [14], EfficientDet [15], Single Shot Detection [16], [17], Faster R-CNN [11], Mask R-CNN [12], and ExtremeNet [18] are available for transfer learning.

EfficientDet [15] is an object detection model developed by Google Brain and belongs to the one-stage object detection algorithms. The focus of this model is on small size and fast inference time. Despite the fast inference time, EfficientDet also demonstrates acceptable accuracy when measured against the Common Objects in Context standard dataset [19]. Models based on the EfficientDet architecture require only a small number of model parameters and floating-point operations, resulting in a compact memory size. EfficientNet [20] serves as the backbone of this architecture. EfficientNet is one of the most powerful convolutional neural network architectures and can be scaled efficiently. Since only a limited computing power and memory size of the model can be used in this paper due to the browser application, EfficientDet is employed to meet this requirement.

[1] https://www.kaggle.com/datasets/farmermatzle/keyboard-key-detection (last accessed March 5, 2024).

Fig. 4. Final result of the implemented key detection model based on the EfficientDet architecture for different test persons

The trained model (see Figure 4) achieves accuracy of over 85% for both average precision and average recall with unknown input data. This clearly shows that the trained model has high accuracy and robustness against various input data.

3.3 Technical realization of hand pose estimation

The ability to detect hand positions and movements is crucial for this task. The hand positions provide information on whether the button was pressed with the correct finger to verify the touch-typing system. Robust hand detection in real-time is a highly complex and demanding computer vision task. For this purpose, an efficient two-stage hand tracking pipeline that detects multiple hands in real time on mobile devices has been developed. This finger tracking solution uses machine learning to recognize 21 3D landmarks from an image. The realized machine learning pipeline consists of two different models. The first model uses the entire input image and provides a box frame for the palms. The reason for this is that the palm is a relatively small object and can be easily modeled using the bounding box method [21].

The second model in this pipeline works with the image section of the hand previously defined by the Palm Detector. Finally, 21 3D hand key points (see Figure 5) are identified and calculated for this image section using the Hand Landmark Detector.

Fig. 5. Final result of the implemented hand pose estimation model for different test persons

3.4 Technical realization of the Typing Learning Academy

In order to implement the key detection and hand pose estimation technology (see Figure 6), the Typing Learning Academy (see Figure 7) was developed, offering a total of 170 learning courses. This learning web application is designed for both beginners and advanced users of touch typing. The difficulty level of the individual courses increases progressively to enhance the development of typing skills. Based on the technical implementation of the Typing Learning Academy, a study was conducted with several participants to analyze the effectiveness of the learning progress.

Fig. 6. Position comparison of the key detection bounding boxes and finger positions of the hand pose estimation

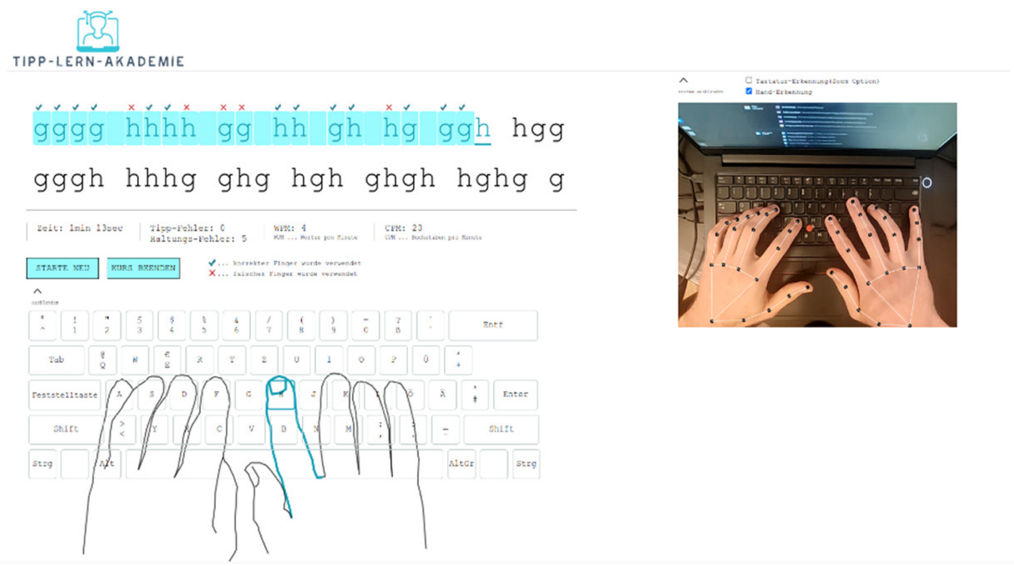

Fig. 7. Typing Learning Academy learning course position control

4 EVALUATION

This study is a moderated test. Tasks are presented in written form as a script for the test participants to complete. This study is a technical and didactic analysis of the Typing Learning Academy. The first task involves a general review of the user experience and usability of the Typing Learning Academy. The second task of this scientific study is to evaluate whether touch-typing can be learned more efficiently and quickly by using the Typing Learning Academy. As the study in this paper is not a long-term study, the participants were specifically asked whether they could envision enhancing their tactile system in the future by regularly using the mobile app. Precise analyses can be derived from the various answers provided by the test subjects.

In order to obtain meaningful results from the study, the test participants must be given predefined tasks. This task was developed to cover and test various elements of the Typing Learning Academy. The test subjects are asked to complete specific tasks to identify errors in the web application. The tasks also aim to analyze the effects of using the Typing Learning Academy platform on learning activities and learning effectiveness.

To conduct a meaningful study and obtain representative results, the selection of test participants plays a crucial role. The test subjects were selected with the aim of obtaining meaningful results and diverse feedback from the study. Care was taken to select test participants who accurately represented the target group. Test subjects with experience in touch typing and test subjects without experience in touch typing were selected. People with technical understanding and people without technical understanding were selected to achieve a certain diversity in the study. To uncover usability problems, testing groups with fewer than ten test subjects are sufficient to detect errors. According to [22], just five test subjects are sufficient to detect up to 75% of usability problems. In the study conducted as part of this paper, ten different tests participants with varying technical requirements, including laptop and mobile phone brands, operating systems, and browsers, performed the assigned tasks.

To analyze the study, questionnaires with specially adapted questions were created. A major advantage of this evaluation method is that data can be systematically collected to obtain various information on topics of interest. Another advantage of questionnaires is that they provide a standardized method for collecting data and comparing it across different test groups. Objectivity is also ensured by these standardized questions, as all test participants adhere to the same evaluation criteria. Questionnaires enable efficient data collection, which can be easily evaluated and analyzed.

4.1 Results of the study

Based on the study, the learning progress and user-friendliness of the Typing Learning Academy can be evaluated. All test participants provided mainly positive feedback on the mobile app and recognized its potential for enhancing touch-typing skills. In terms of usability, the structured organization of the learning courses and the various levels of difficulty received positive ratings. Despite having different levels of experience in touch typing, participants can effectively imagine improving their typing skills. By directly controlling the finger position using deep learning methods and the camera stream of the mobile phone, test users paid particular attention to achieving the correct hand position. During the study, it was observed that correct finger positioning led to a reduction in error rate and a significant increase in writing efficiency. The direct finger position feedback from the mobile application also contributed to more effective learning. Users made more accurate course entries with a lower error rate at the end of the study.

The state-of-the-art deep learning technologies used also present some challenges in the current state of this paper. In the case of hand position estimation and key detection, the challenge also lies in detecting positions under various lighting conditions and at different speeds. Specific finger movements, especially when fingers overlap during fast typing, can also lead to limitations in this system and trigger false detections. Another performance issue related to very fast typing may stem from the individual hardware limitations of the test users. For accurate object detection algorithms, it is always necessary to strike a good balance between precision and speed. Faster architectures can lead to lower detection accuracy, but they are better suited for real-time applications.

The conclusion of the study is that the Typing Learning Academy enables effective improvement of typing skills through touch typing. The usability and technical functionality of the web app met the requirements of the test participants and worked smoothly. Furthermore, it can be stated that the results of the study not only reflect the current status of the Typing Learning Academy but also provide insights into technical challenges and potential improvements for this program.

5 DISCUSSION

5.1 Implementation

Based on the implemented object detection and hand pose estimation technology, a web application was developed. This learning web application is designed to simplify the user's interaction with the underlying technologies. A total of 170 learning courses with varying levels of difficulty were developed to enhance the learning progress. By comparing the key positions with the finger positions, it is possible to verify whether the correct fingers are being used for the corresponding keys according to the rules of the touch system. The Typing Learning Academy developed in this paper, with additional verification of correct hand position, is the first of its kind and offers a significant advantage compared to other learning apps. The potential impact of this work is a first-of-its-kind piece of software for real-time finger and keyboard position detection within a learning web application for touch typing. With the help of the test participants, the tracking of the finger position could be analyzed more precisely. Only a few incorrect finger position detections were identified by the individual test participants, indicating that the hand pose estimation technology used is highly robust. As part of the test study, it was also possible to analyze that the keyboard detection also achieves a very high accuracy rate. In this paper, it is shown that hand pose estimation and keyboard detection have only small error deviations, which is also confirmed by the conducted study. In summary, the presented implementation of this paper effectively addresses the research question: "Is it feasible to utilize hand pose estimation technologies and object detection technologies to validate accurate finger placement (touch typing) within a browser environment, in real-time?"

5.2 Didactics

The aim of the scientific study is to analyze the effects of using the Typing Learning Academy platform on learning activity and effectiveness. The study shows that all test participants can envision using the learning app more frequently in the future to enhance their writing skills. The usability and technical functionality of the web app met the requirements of the test participants and worked without any issues. It was shown that the test participants paid more attention to improving their finger posture when they noticed a faulty hand position. As a result, the finger posture also became more correct, and the rate of typing errors was significantly reduced during the test study. These observations were also confirmed by the individual test participants. With the help of the Typing Learning Academy, users are explicitly made aware of incorrect finger posture so that this method can demonstrably achieve more efficient learning progress. These results indicate that the developed mobile learning technology will be successful in achieving a significant improvement in touch typing over an extended period of time. Therefore, the research question is: "Can the learning progress in acquiring touch-typing be enhanced more efficiently through the use of a web application that provides real-time feedback on finger positioning on the keyboard?" This question can also be answered positively.

6 CONCLUSION

This scientific paper presents a practical approach to enhancing touch-typing skills using mobile technologies and deep learning algorithms. This approach has a significant impact on pedagogy and typing efficiency. For this purpose, the mobile web application Typing Learning Academy has been developed, which can significantly improve typing efficiency and speed. The key distinction from other mobile technologies is that the keyboard layout is sent to the learning app through a mobile phone, providing direct feedback on which key to press with each finger. The key and finger positions are determined using deep learning technologies to monitor the user's typing posture in real time. Direct feedback on finger position ensures more efficient learning and improved usability. Mobile technologies, combined with deep learning technologies, make a significant pedagogical contribution to enhancing the efficiency of learning specific skills, as shown in detail in this paper.

7 REFERENCES

[1] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018. https://doi.org/10.1016/j.patcog.2017.10.013

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

[3] L. Domingo and F. Borondo, "Deep learning methods for the computation of vibrational wavefunctions," *Communications in Nonlinear Science and Numerical Simulation*, vol. 103, p. 105989, 2021. https://doi.org/10.1016/j.cnsns.2021.105989

[4] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," https://proceedings.neurips.cc/paper_files/paper/2013/file/f7cade80b7cc92b991cf4d2806d6bd78-Paper.pdf

[5] B. Doosti, "Hand pose estimation: A survey," *arXiv*, 2019. https://doi.org/10.48550/arXiv.1903.01013

[6] A. Ikhsananto and S. Sutirman, "The effectiveness of rapid typing software and module for improving 10-finger typing skills," *Dinamika Pendidikan*, vol. 13, no. 2, pp. 228–237, 2018. https://doi.org/10.15294/dp.v13i2.15540

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv*, 2015. https://doi.org/10.48550/arXiv.1506.02640
[8] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *Computer Vision – ECCV 2016*, vol. 9905, pp. 21–37, 2016. https://doi.org/10.1007/978-3-319-46448-0_2
[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," arXiv:1708.02002 [cs], 2018. https://arxiv.org/abs/1708.0200
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv*, 2014. https://doi.org/10.48550/arXiv.1311.2524
[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv*, 2016. https://doi.org/10.48550/arXiv.1506.01497
[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv*, 2018. https://doi.org/10.48550/arXiv.1703.06870
[13] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv*, 2017. https://doi.org/10.48550/arXiv.1611.10012
[14] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," *arXiv*, 2019. https://doi.org/10.48550/arXiv.1904.08189
[15] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," *arXiv*, 2020. https://doi.org/10.48550/arXiv.1911.09070
[16] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv*, 2017. https://doi.org/10.48550/arxiv.1704.04861
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv*, 2015. https://doi.org/10.48550/arXiv.1512.03385
[18] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," *arXiv*, 2019. https://doi.org/10.48550/arXiv.1901.08043
[19] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," *arXiv*, 2014. https://doi.org/10.48550/arXiv.1405.0312
[20] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *arXiv*, 2020. https://doi.org/10.48550/arXiv.1905.11946
[21] A. Alvin, N. Shabrina, A. Ryo, and E. Christian, "Hand gesture detection for American sign language using k-nearest neighbor with mediapipe," *Ultima Computing: Jurnal Sistem Komputer*, vol. 13, no. 2, pp. 57–62 2021. https://doi.org/10.31937/sk.v13i2.2109
[22] J. Nielsen, "Usability engineering," *Elsevier*, 1994. https://books.google.at/books?hl=en&lr=&id=95As2OF67f0C&oi=fnd&pg=PR9&dq=NIELSEN

8 AUTHORS

Mathias Mattersberger is a student of Masters in Computer Science at Graz University of Technology. His interests include visual computing and deep learning technology (E-mail: mathiasmattersberger@gmx.at).

Josef Wachtler is currently working as an Edtech developer at the Department of Educational Technology at Graz University of Technology. Furthermore, he holds a PhD in computer science and assists in supervising Master's and Bachelor's theses. His research-interests lie in the field of video-based learning, applied in various settings such as schools, universities, or MOOCs (E-mail: josef.wachtler@tugraz.at).

Martin Ebner is currently the Head of the Department of Educational Technology at Graz University of Technology. He is responsible for all university-wide e-learning activities and serves as a Senior Researcher in educational technology (E-mail: mebner@gmx.at).