EPiC Series in Computing
Volume 107, 2025, Pages 146–155
Proceedings of EUNIS 2025 annual congress in Belfast

Beyond the Hype: Real-World Insights on AI Integration and the Value of Alternative Solutions

Hanbit Choi[1], Aksel Skullerud Fosaas[1], Zara Mudassar[1], Sander Garland Slinning[1], Arman Bulak Hagelia[1] and Eira Bjerkeng Scherjon[1]
[1] Sikt - Norwegian Agency for Shared Services in Education and Research, Norway
hanbit.choi@sikt.no, aksel.fosaas@sikt.no, zara.mudassar@sikt.no, sander.slinning@sikt.no, arman.hagelia@sikt.no, eira.schjeron@sikt.no

Abstract

This paper discusses the application of artificial intelligence (AI) in two distinct projects in Norway's higher education sector. Team SAIL aimed to develop an AI-driven solution for document classification, addressing the time-consuming challenges faced by admission officers. The team created two solutions: one using ResNet, a pre-trained image classification model with transfer learning, and another client-side solution that classifies documents based on relevant text. The client-side solution was developed as a direct response to GDPR restrictions encountered during the development of the first model. Both solutions showed promising results, however implementation issues are present due to strict regulations such as GDPR.

Team Loggerne aimed to apply machine learning to the problem of analyzing and connecting logs from multiple different sources. They explored different ML techniques, ultimately deciding that the data was not fit for ML, and that the best option was to move away from ML and attempt other approaches.

This paper discusses both teams' experiences, illustrating that while AI can be effectively used to improve efficiency, it might not always be suitable.

1 Introduction

In the summer of 2024, two interdisciplinary teams composed of students, including software developers and designers, came together to tackle unique challenges that were heavily focused on artificial intelligence (AI). These teams, named Team SAIL and Team Loggerne, were each assigned distinct problem statements that inherently emphasized the application of AI solutions. This article discusses the specific problems faced by each team, with regards to approaches and outcomes. These case studies highlight an important insight: while AI can be a powerful tool, it is not always the optimal solution to every problem. This article provides a nuanced view on AI, discussing complexities of integrating AI into real-world applications and the value of considering alternative approaches when appropriate.

1.1 Team SAIL's problem

Team SAIL was tasked with utilizing AI for document classification problems faced by admission officers in the higher education application process. Admission officers are responsible for reviewing a wide variety of documents submitted by applicants. However, these documents often come with non-descriptive file names, typically consisting of random letters and numbers, which makes it difficult to quickly identify and categorize them. Adding to the complexity, some documents, such as police certificates of conduct, are classified as sensitive and shall not be accessed by the admission officers at all.

This results in a time-consuming and inefficient workflow for admission officers, who manually sift through numerous documents to determine their relevance and sensitivity. Team SAIL's objective was to automate this process by implementing an AI-driven solution.

1.2 Team Loggerne's problem

Throughout their life, most people will interact with different parts of the education sector's ICT systems spanning multiple years. This generates a large number of logs, detailing a user's experience. Such logs can tell a lot about a system, and having the ability to understand and connect them brings great value. Sikt provides students and institutions with multiple services within higher education, such as admissions, diploma registry, and course administration. The large number of logs generated by these services are difficult to analyze due to each system's differences in log-structure and logging practices.

Team Loggerne was tasked with finding patterns in these logs; while also attempting to connect the different types of logs users have generated across Sikt's services, to find problems that needed to be addressed. Their initial goal was to explore how machine learning (ML) could be used in the process of analyzing the logs.

2 Approach

This section explores various methods for leveraging AI in the project. The first approach involves document classification using machine learning, addressing challenges related to limited data and GDPR restrictions. An alternative client-side classification method using Wasm is then introduced, emphasizing privacy and efficiency. Lastly, the application of AI for log analysis is examined, revealing key limitations that led to a reassessment of machine learning's suitability for the task.

2.1 Document Classification Using Machine Learning

Classification of documents using machine learning requires multiple steps. These steps include collecting data, preprocessing it, defining the model architecture and pipeline for training, experimenting and finding the most optimal values for hyperparameters, and gathering results.

2.1.1. Dataset

The documents uploaded to the Norwegian university admission systems often contain sensitive or personal information. To train a machine learning model on these documents following the GDPR, they have to be anonymized making it impossible to identify who the document belongs to (Biesner et al., 2021). To obtain these documents in compliance with GDPR, a legal expert was consulted, and the privacy policy was updated, allowing the team to manually anonymize a small set of documents. Following this, a custom dataset containing 10-20 samples for each document type was set together.

To increase the sample size, each document was augmented to generate 20 variations, enhancing both data quantity and diversity in terms of blurriness, lighting, and rotation. The data was split into training, validation, and test sets before augmentation to prevent data leakage.

The dataset was divided into seven classes, such as certificate of competence, military service proof, police certificate, and foreign diplomas.

2.1.2. Model Architecture and Pipeline for Training

The model was trained using Transfer learning, where a pre-trained model was fine-tuned for a related task. This approach leveraged prior knowledge to improve performance, especially with limited data. Training occurred in two stages: pre-training on a large dataset to learn general patterns, followed by fine-tuning on a smaller, task-specific dataset (Shaha & Pawar, 2018).

ResNet (He et al., 2016), pre-trained on ImageNet (Deng et al., 2009), was used as the base model. Fine-tuning occurred in two stages: first, all layers except the final classification layer were frozen, allowing the model to adjust to seven document classes. A dropout layer was added to reduce overfitting. In the second stage, the entire model was unfrozen and trained on the dataset, enabling accurate classification of document types.

2.1.3. Experiments

Experiments followed a trial-and-error approach to optimize hyperparameters and data augmentation. ResNet50 was used as the base model with a dropout of 0.8. Learning rates were set to 0.001 and 0.0005 for the two fine-tuning stages to prevent overfitting. The model was trained for 20 epochs in the first stage and 30 in the second, achieving optimal performance.

Balancing augmentation was crucial for generalization. Too many augmented documents reduce diversity, while too few causes overfitting. After experimentation, 20 augmented samples per document were chosen as optimal.

2.1.4. Results

The machine learning model was evaluated on a test set containing 93 documents previously unseen by the model. The accuracy for the model on this test set was 0.94. Figure 1 shows the confusion matrix for the model when evaluated on the test set.

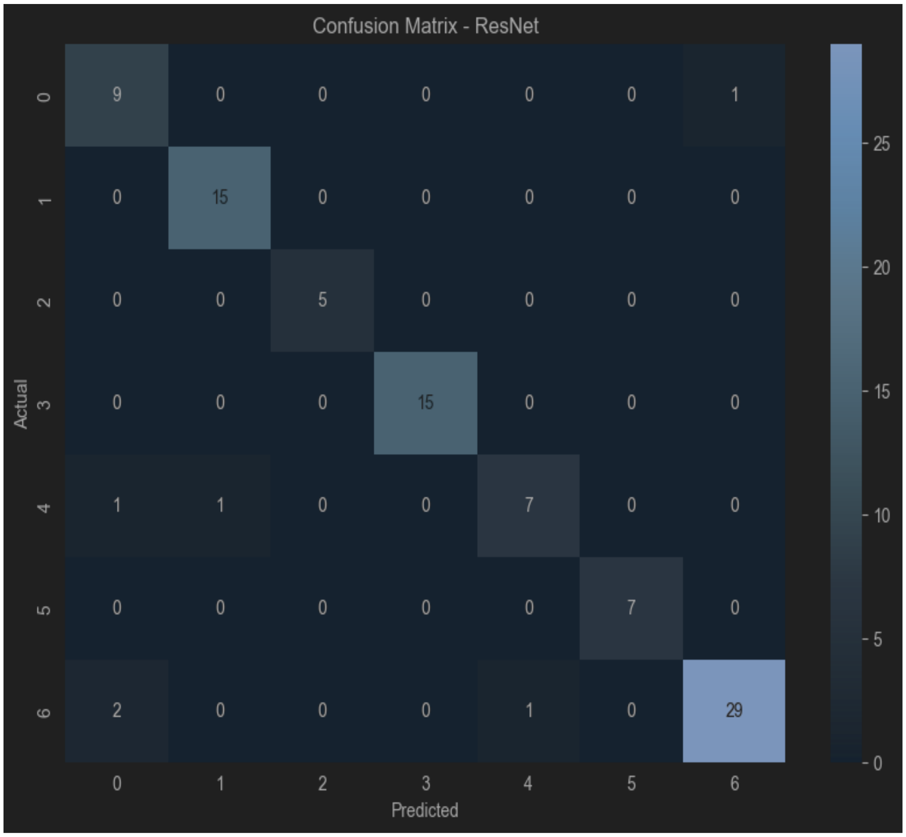


Figure 1: Confusion matrix for the AI model when evaluated on the test set.

2.2 Client-side Classification

After issues with GDPR regarding the machine learning model were discovered, a simpler, client-side solution using Wasm (WebAssembly) was explored. Wasm is a binary instruction format for a stack-based virtual machine (Haas et al., 2017). In practice, this means that developers can write code in languages like C, C++, or Rust, and compile it into a compact, fast-loading binary format that runs directly in the browser at near-native speed. This allows the browser to execute more complex tasks than what a browser typically does, such as running machine learning models or parsing documents.

2.2.1. Process

The Wasm-based classification was essentially built on top of Tesseract.js, an existing Wasm-based library for extracting text from images (Tesseract.js, n.d.). The first step was to make tessaract.js scan the image of the given document. Then after getting the text from the document, it was compared to one or several arrays of predefined words that could identify a specific document. For example, for military service proof the array could be [“national service”, “proof of service”, “military”]. Every word in the array is searched for in the text extract from the image. Due to the unreliability of Tessaract.js, the Sørensen-Dice coefficient (Dice, 1945) is used to compare the similarity between the words in the document and the words in the array. If the similarity is above a given threshold, the word in the array is considered found in the document. If all words in the array are found in the text extract from the tesseract.js scan, the document is classified as the document type corresponding to the given array.

2.2.2. Results

From the testing that was done, the Wasm-based classification seemed to produce good results. However, no data was collected to determine exactly how accurate the classification was. Additionally, the parameters can always be tweaked to favor the training set, and the actual results on unseen data are unknown. As time passes, the wording in documents is likely to change, possibly reducing accuracy over time.

2.3 Team Loggerne's Approach

Team Loggerne explored two main machine learning approaches when trying to analyze logs from Sikt's services. The initial approach, which was feature-based, was decided based on the team's experience and discussions with other developers at Sikt. After some analysis, the team understood that this approach would be difficult as the large number of features in the dataset made it hard to find any real patterns in the logs.

For the second approach the team restructured the logs to be divided into sessions for each user. The plan was to use the sequences of actions to find patterns in the logs with ML, however the given data were not as fitting for ML as the team had hoped. After dividing the logs into sessions, it was found that for sessions with three actions, 20% were completely unique, and for sequences with 10 user-actions, 67% were unique. The uniqueness of the sequences would cause issues, but the team still had hope that the remaining sequences were concentrated over a few different combinations. After reviewing the remaining sequences however, it was found that there were no specific sequences that were significantly more common than others. In light of these findings, the team decided not to go with a machine learning approach for analyzing the logs and explored other solutions.

3 Discussion

3.1 Strengths and Weaknesses of the Machine Learning Model for Classification

The machine learning model offered several advantages and accurately classified low-quality document photos. By automating the classification process, the manual workload was reduced, and human errors were minimized. Additionally, transfer learning allowed the model to adapt effectively despite limited training data, making it a viable solution while ensuring GDPR compliance.

The machine learning model also had several disadvantages. Limited training data due to GDPR restrictions hindered performance improvements, and while augmentation helped, it could not replace a diverse dataset. Another issue regarding GDPR was the difficulties of getting permissions to use documents in machine learning, making the implementation process difficult. High computational costs made training and fine-tuning challenging, requiring powerful GPUs or cloud solutions. Additionally, real-world variations in document layouts, fonts, and languages could reduce classification accuracy, necessitating continuous updates and retraining.

3.2 Advantages and Future Possibilities of Client-side Classification

The biggest advantage of using client-side classification was privacy. With Wasm-based classification, all the data processing happens client-side. This means that if an applicant tries to upload sensitive information that the application processor is not supposed to see, it can be stopped before the sensitive information is sent to the server.

While the type of classification in this case uses OCR (Optical Character Recognition) and string similarity, there are other ways to utilize Wasm to do client-side classification. With the rise of LLMs in recent years, LLMs have also made their way into the browser with the power of Wasm. WebLLM is a high-performance in-browser LLM inference engine facilitating the use of LLMs directly in the browser (MLC Community, 2024). In its current state WebLLM seems unsuitable for the task at hand because it requires downloading the model taking a lot of time, and its performance is not as strong compared to running models on servers like OpenAI's. However, exploring the future possibilities with WebLLM is suggested.

Another powerful tool for client-side classification is TensorFlow.js, a JavaScript library that enables the deployment of machine learning models directly in the browser. (TensorFlow, n.d.) With TensorFlow.js, developers can run pre-trained models or train new models right in the browser, providing an efficient way to classify data without needing to send sensitive information to a server. This opens up significant possibilities for real-time, privacy-preserving applications. While TensorFlow.js models may not always match the performance of server-side models due to browser limitations, the library is constantly evolving, and with the advent of optimizations such as WebGL acceleration, it offers increasingly competitive performance. As TensorFlow.js continues to grow, it holds great potential for developing secure, high-performance applications that leverage machine learning without compromising privacy.

3.3 Limitations of Client-side Classification

Using the Wasm-based classification in its current state has some obvious disadvantages. For one, to make the array of words to look for in the uploaded document, one has to already be aware of some of the expected content. While this is not an issue for documents that have a clear structure and have uniquely identifying words, other documents might be affected by this issue. This is also a problem with structured documents if the documents change format over time.

Additionally, there is the problem of image quality. In higher education admissions the image quality of the uploaded document tends to vary. Tesseract.js requires images to be of a certain quality to be able to read the text. One solution to this problem could be to ensure that images are of a certain quality.

Another limitation of client-side classification is the performance of the client's computer. While Wasm allows the utilization more of the power of the computer, it is still limited by the performance of the machine itself. Applicants for higher education, which typically are high-school students, usually do not own the most powerful computers and it is also possible to apply using a smartphone which has even less power. This makes a client-side solution more challenging.

3.4 Team Loggerne - Discussion

While Team SAIL found that AI was useful for solving their tasks, team Loggerne had another experience. Their objective was to analyze and establish connections between logs from Sikt's various systems to enhance educational services. AI and machine learning were explored as potential tools for identifying patterns and insights within the provided log data, with the intention of presenting findings through an application for employees or users. However, further analysis revealed that machine learning might not be the most suitable approach. Despite initial expectations that an ML-based method would yield meaningful insights, analysis exposed significant limitations. The log data exhibited high uniqueness and lacked recurring patterns, making generalization challenging for ML models. Various feature reduction techniques were applied to manage the large number of dataset features, but the results were insufficient.

A thorough evaluation of the data ultimately indicated that machine learning was not an appropriate solution for this problem. This outcome highlights the importance of assessing data carefully before applying ML to ensure its effectiveness. Machine learning should only be utilized when it provides clear advantages over conventional methods in terms of efficiency, scalability, and predictive power.

3.5 Is AI always necessary?

Machine learning is a powerful tool for analyzing data and making predictions, but it is not always the best approach. It is most effective when applied to large and complex datasets that contain clear patterns, allowing for the development of reliable models capable of making accurate predictions. In the case of Norwegian university admission systems, a machine learning model effectively processes and categorizes documents such as military service proof and foreign diplomas, despite challenges like low-quality images and data constraints. By using transfer learning with a pre-trained model like ResNet, the system adapts to the unique nature of the data and significantly improves classification accuracy. This demonstrates that when the dataset is large and complex, ML offers a robust solution, automating the process and improving both efficiency and accuracy.

However, ML is not suitable when data is sparse, highly unique, or lacks discernible patterns, as models may struggle to generalize effectively. A clear example of this is seen in the attempt to use ML for analyzing log data from Sikt's various systems. Despite initial expectations, the log data exhibited high uniqueness and lacked recurring patterns, which made it difficult for ML models to generalize effectively. Various feature reduction techniques were applied, but the results were insufficient. This case highlights that ML is not always the best tool, especially when the data doesn't contain clear patterns or enough consistency to support reliable predictions. It emphasizes the need to assess the nature of the data carefully before deciding to apply machine learning, as simpler methods may be more effective in certain situations.

Additionally, ML introduces problems related to data privacy and GDPR making it challenging to implement these solutions into real world applications. During the process of developing all the different solutions, assistance from the legal team was needed at several stages. Often times it was not clear what was okay and what was not okay in terms of dealing with data privacy.

4 Conclusion

This study highlights both the potential and limitations of AI in university admissions processes. While SAIL successfully implemented an AI model for document classification, privacy concerns necessitated the alternative approach of using Wasm instead. Loggerne's work with log data analysis revealed that traditional methods were more effective than AI due to data uniqueness challenges. These findings emphasize the importance of evaluating AI's suitability for specific tasks rather than assuming it is always the best solution. AI should be used as a means to reach a goal and not be a goal in itself.

References

Biesner, D., Ramamurthy, R., Stenzel, R., Lübbering, M., Hillebrand, L., Ladi, A., Pielka, M., Loitz, R., Bauckhage, C., & Sifa, R. (2021). Anonymization of German financial documents using neural network-based language models with contextual word representations. International Journal of Data Science and Analytics, 13(2), 151–161. https://doi.org/10.1007/s41060-021-00285-x

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, 26(3), 297–302. https://doi.org/10.2307/1932409

Haas, A., Rossberg, A., Schuff, D. L., Titzer, B. L., Holman, M., Gohman, D., Wagner, L., Zakai, A., & Bastien, J. (2017). Bringing the web up to speed with WebAssembly. Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation. https://doi.org/10.1145/3062341.3062363

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).

Shaha, M., & Pawar, M. (2018, March). Transfer learning for image classification. In 2018 second international conference on electronics, communication and aerospace technology (ICECA) (pp. 656-660). IEEE.

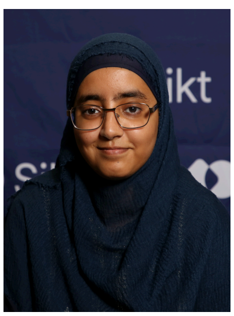
TensorFlow. (n.d.). TensorFlow.js. Retrieved February 21, 2025, from https://www.tensorflow.org/js

Tesseract.js. (n.d.). Tesseract.js: Pure JavaScript OCR for 100+ languages. Project Naptha. Retrieved February 21, 2025, from http://tesseract.projectnaptha.com

Author biographies

Hanbit Choi is in the final year of her bachelor's degree in software engineering at Oslo Metropolitan University. She has been working as a developer at Sikt since last summer, starting as a summer intern and later continuing as a part-time developer. During her internship, she focused on data analysis and designing optimized data structures for future use. Since last autumn, she has been part of Team Sail and continues to work with them. In addition to her role at Sikt, she is collaborating with Fürst on her bachelor's thesis, where she is developing a digital dictation application.

Aksel Skullerud Fosaas is currently in the final year of his bachelor's degree in Informatics at the Norwegian University of Science and Technology (NTNU) in Trondheim. He started working as a developer at Sikt for an internship last summer and has worked part time since. During the summer he worked closely with 5 other students doing data analysis and similar work with the objective of improving logging practices within Sikt's services. Following the summer, he spent his time as a part of Team SAIL.



Zara Mudassar is in her last semester at the Oslo Metropolitan University (OsloMet), pursuing a master's degree in informatics with Applied Artificial Intelligence as her specialization. She has acquired significant professional experience at Sikt – the Norwegian Agency for Shared Services in Education and Research – through two full-time summer positions and ongoing part-time work throughout the year. While working at Sikt, she has been part of a student team working with document handling using AI. She, together with the team, has explored document anonymization, synthetization, information extraction, as well as classifying documents relevant to Norwegian university applications.

Sander Garland Slinning is currently pursuing a master's degree in informatics at the Norwegian University of Science and Technology (NTNU), where he also holds a bachelor's degree in the same field. He has gained valuable professional experience through his work at Sikt – the Norwegian Agency for Shared Services in Education and Research, where he worked both part-time during the semester and full-time over two summers. During his time at Sikt, Slinning contributed to the development of the new Student ID App and is now a member of Team SAIL, where he collaborates with the team to explore automated document classification solutions for university admissions.

Arman Bulak Hagelia is currently working as Team Lead for the AI/ML team and a service designer at Sikt. He previously pursued a master's degree in informatics: Design, Use, Interaction at the University of Oslo. His thesis explores multidisciplinary approaches to project portfolio management as a means to successful digitalization in the Norwegian higher education sector.

Eira Bjerkeng Scherjon is currently writing her master's thesis in Interaction Design at The Oslo school of Architecture and Design. Through two full-time summer positions and ongoing part-time work she has worked in a student team with document handling using AI. Together with four developers, she has explored the design aspect of document anonymization, synthetization, information extraction, as well as designing a frontend solution for stopping irrelevant documents uploaded with applications for higher education in Norway.