INSTITUTIONEN FÖR DATAVETENSKAP | LUNDS TEKNISKA HÖGSKOLA | PRESENTERAD 2024-03-14

EXAMENSARBETE Management of Training Data for Deep Learning Applications: Requirements and Solutions
STUDENTER Adla Lagström Jebara, Fabian Sundholm
HANDLEDARE Lars Bendix (LTH)
EXAMINATOR Emelie Engström (LTH)

Uncharted Territories: Exploring Data Management in Software Development

POPULÄRVETENSKAPLIG SAMMANFATTNING **Adla Lagström Jebara, Fabian Sundholm**

Within software development, the management of source code has been extensively studied. The same level of attention, however, has not been devoted to the management of associated data, such as training data used in machine learning algorithms.

In recent years, deep learning technologies have advanced rapidly, which has resulted in the launch of several new applications such as image generating models and chatbots. These applications are trained on large datasets to recognize patterns and make predictions or classifications based on new data they encounter. However, despite deep learning being almost a household name at this point, very little is known about how the companies developing these applications manage the vast amount of data that is required to create these applications. There is also very little public research available on data management in a machine learning context.

At the case company where this research was conducted, machine learning algorithms are used in combination with other technologies, such as conventional image analysis methods, to provide tailored fingerprint recognition solutions, i.e. technology that identifies and verifies individuals based on their unique fingerprint patterns. However, the process of working with the vast amount of data have been described by developers at the company as "frustrating", "complex", and "quite messy", and suffers from numerous challenges.

To address this problem, our research aimed to investigate if there is a scalable solution for storing and managing training data, and thereby enhance the effectiveness of the developers.

The research included identifying the system's needs and how it should work, developing designs to meet these needs, and then testing out one of these designs to see how it works.

To identify the system's needs, we did a literature study and interviewed several developers at the case company. To come up with design solutions, we looked at available tools to determine if any existing tool could fulfill the system's needs. We found two different design solutions, and one of them was selected for implementation as a proof of concept.

However, we found that the proposed design solution did not fully meet the system's needs, indicating a complexity in addressing the problem beyond our initial expectations. Moreover, we found that an available tool on the market fulfilling all the system's needs to a satisfactory degree likely does not exist. Nevertheless, our research indicates that with additional time and resources, it is feasible to address the problem and develop such a solution by implementing the identified needs of the system, which we be consider to be an interesting area for future work.