In recent years, Convolutional Neural Networks (CNNs) have gained immense popularity for their ability to deliver satisfactory accuracy in various machine learning and computer vision applications. The landscape is rich with a wide array of CNNs, each differing significantly in terms of execution time, energy consumption, and inference accuracy. To accelerate CNN execution, several hardware accelerators have been introduced, with Xilinx's Deep Learning Processor Unit (DPU) being a notable example. The DPU offers a multitude of configuration options and is capable of executing a diverse range of CNNs. However, the choice of CNNs and the configuration options in these accelerators create a vast design space, necessitating detailed trade-off analysis to design efficient systems. To support such analysis, this thesis introduces several variations to a framework for estimating the execution time and energy consumption of CNNs when executed on DPU-based systems. This framework offers invaluable assistance to system designers in selecting accelerator and CNN configuration parameters based on specific application requirements, ultimately leading to optimized performance.

Moreover, the methodologies proposed in this thesis are generic and can be applied to a wide range of CNN accelerators. The thesis begins with the introduction of INFER, a methodology designed to estimate the execution time of any CNN on a given DPU size without the need for actual implementation. INFER operates within restricted DPU configurations, allowing up to only three DPUs active concurrently and employing the default bus interconnections provided by the Xilinx tool to the external memory. Additionally, it can estimate the additional time required due to memory interference resulting from the concurrent use of multiple DPUs. Energy estimation for CNNs running on a DPU is another focus of this thesis. An energy estimation technique named EnergyNN, is developed by leveraging the characteristics of CNNs and DPUs. This technique can be applied to predict the energy consumption of even newer CNNs not used in the model development. Extensive evaluation demonstrates the effectiveness of these approaches, with average prediction errors of 6.6% for execution time and 8.8% for energy estimation. The utility of these predictions is further illustrated through real-world applications in traffic monitoring and drone systems.

The thesis also expands its scope to explore the design space consisting of a larger number of DPUs, thus effectively increasing the concurrency. However, this expansion introduces complexities related to bus connections and CPU core limitations. To address these challenges, the thesis introduces an execution time predictor designed to optimize DPU configurations to meet the performance demands of diverse tasks. Various methods are employed to isolate concurrent CNNs from each other and the operating system. A machine learning-based prediction approach named EXPRESS, is then introduced to predict the execution time of a given CNN on a DPU configuration, considering the characteristics of the CNN, DPU, and bus. EXPRESS provides predictions for both CPU and DPU processing times, resulting in the estimation of end-to-end processing times. To further enhance this approach, EXPRESS-2.0 is introduced, offering support for heterogeneous CNNs. To avoid having different number of features for different count of DPUs/CNNs, EXPRESS-2.0 consolidates the features of Co-runners (CNNs which run concurrently with the CNN for which we predict the execution time) together.

A controller has been developed to isolate CPU cores responsible for executing the operating system and the actual CNNs. This isolation reduces the variation in execution time measurements and enhances prediction accuracy. Across all these methodologies, machine learning based regression techniques are employed for prediction. All experiments are conducted on a real FPGA board, and the evaluation, featuring 16 standard CNNs, reveals very low prediction errors. EXPRESS and EXPRESS-2.0 achieve an average prediction error of 2.2% and 0.7%, respectively. The low prediction error of the framework make it highly effective for design space exploration, offering significant benefits to embedded system application developers.