Problem Understanding

- Industrial Context

The process of making semiconductors is very precise. Has many steps that have to be done just right. Semiconductor fabrication has a lot of chemical, thermal and lithographic steps that have to be controlled carefully. If there is a tiny problem with the semiconductor fabrication like a scratch on the surface or a pattern that is not right it can cause big problems with the whole thing. Semiconductor fabrication can also be messed up by chemicals. Any little flaw in the semiconductor fabrication can make the whole semiconductor fabrication process fail or not work properly which is a deal.

- Challenges in Current Inspection Systems

Traditional quality control in factories is dealing with a big problem. They are getting too much data. Every day they get terabytes of pictures from inspections. These pictures are taken with machines, like SEM and Optical. The old systems they use to check these pictures are not working well. They have trouble with:

When we talk about High Latency it is really about delays. These delays happen because we use analysis. This means that all the information goes to one place. We also use review stations. These stations are like roadblocks. They stop us from making changes, to the process away. With High Latency we cannot make changes in time. This is a problem. High Latency is an issue that affects how we work with analysis and manual review stations.

The cost of keeping the internet connections and computer servers that are needed for cloud-based analysis is really high. This is what we call Infrastructure Overhead. The thing is, we need fast internet connections and strong computer servers to do this kind of analysis in the cloud.. It is just too expensive to keep all of this up and running. Infrastructure Overhead is a problem because of this.

Data sovereignty is really important. When we are talking about Data Sovereignty we have to think about the safety of our information. In the industry that makes these computer chips the designs are very secret and valuable. So when we send the data from checking these designs to a cloud that is a big risk. The cloud is not a place for Data Sovereignty. We are talking about Data Sovereignty and Data Sovereignty is crucial for our industry.

- Significance of Edge-AI

Edge-AI makes it possible for devices to check for defects away with almost no delay. This means devices do not have to rely much on internet connections to work properly. When devices can think for themselves at the spot where the information is collected, companies that make things can get results quickly and manage their production better. This is what Industry 4.0 standards are all about and Edge-AI helps them do it. Edge-AI is really good for manufacturers because it helps them with Edge-AI and makes their work easier.

Proposed Approach

- Key Concept: TCM-Resident Inference

Our basic way of doing things is Hardware-Software Co-Design. We did not use a model. Instead we built our solution for the NXP i.MX RT series architecture. The main idea is to make sure the model is inside the

processor's Tightly Coupled Memory. This means the model runs in a way without waiting and it does not have the delays that usually happen with external Flash or DDR memory. The NXP i.MX RT series architecture is what our solution is made for and the Hardware-Software Co-Design is how we do it.

- Technical Methodology

We picked MobileNetV2, the one with Alpha 0.35 as the part of our system that helps us understand what is in the pictures. This is because MobileNetV2 is really good at finding the things in a picture without needing a lot of extra information. MobileNetV2 is a choice for us because it is good at what it does and it does not need a lot of space. We like MobileNetV2 for our Lightweight Backbone.

The resolution was made better by using a size of 160x160. This helps to make the file size a lot smaller while still keeping the details we need to see the defects with a microscope. We need to see these details to find the defects. The 160x160 resolution is good for detecting defects.

We had to make some changes to the graph headers. The toolchain was causing problems with metadata locks. So we used Python to fix the ONNX graph headers. This was like a surgery on Graph . We did this to make sure the ONNX graph headers worked with our hardware-optimized shape targets. We had to do this to make the graph headers and our hardware work well together.

Dataset Plan

Our team used a set of 5,000 images from Kaggle. We picked this set of already labeled pictures because it was better than labeling a small set ourselves. This way we got a base for our model so it can handle the differences in sensor noise and lighting on wafers that happen in factories.

By using pictures that were already labeled we were able to focus our engineering work on important things like making sure the model works well on the hardware and making it run faster with INT8 quantization. We also worked on optimizing the model for deployment.

- Classification Schema

The information is divided into eight groups, which is what the India Electronics and Semiconductor Association (IESA) needs.

There are Six Defect Types. These Six Defect Types include some bad problems like cracks, in the material shorts that can cause a lot of damage bridges that form between things they should not be connected to and contamination that can ruin everything.

We have two kinds of classes. The first one is called "Clean". This is for wafers that pass inspection. The other class is called "Other". This is, for things that are not classified and do not look right.

Note About Efficiency: We chose to use image-level classification of manually drawing boxes around things in pictures. This made our model a lot smaller; it is now 627.7 KB. This is really important because we need to make sure things happen fast in less than a millisecond on smaller devices, like edge hardware.

- Preprocessing and Augmentation

We made sure all the pictures were the size so the computer program would work properly and use the right amount of memory. This is what we call Dimensional Standardization. We did this to keep the model stable

and make sure it always uses the amount of memory. This way the Dimensional Standardization helps the computer to work better with the images.

When we get ready, for quantization we need to make sure the pixel values are just right. So we scale these values to a range. This is also called normalization. We do this to keep the math stable when we use INT8 quantization. Quantization is a part of this process and scaling the pixel values helps with quantization.

Industrial Alignment (Channel Broadcasting): We developed a specialized input pipeline to align industrial SEM (Scanning Electron Microscope) data with our neural network backbone. While the microscope provides 1-channel grayscale data, the optimized backbone requires a 3-channel input. We implemented a broadcasting strategy that replicates the grayscale channel across three dimensions, satisfying the model's structural requirements while preserving the integrity of the original industrial feature data.

Model Plan and Optimization

- Development and Training

We used some tools that work with Python like TensorFlow and PyTorch to make our model.
These tools were really helpful because they let us use something called transfer learning.
Transfer learning is when we take a model that is already trained on some things and we change it so it can work with the textures that are on semiconductor wafer surfaces.We did this so the model can really understand semiconductor wafer surfaces.The semiconductor wafer surfaces have some unique textures that the model needs to learn about.We wanted the model to know all about semiconductor wafer surfaces so it can do its job well.The textures, on semiconductor wafer surfaces are important for the model to learn.

- Optimization for the Edge (NXP eIQ)

So we wanted to take our research model and turn it into something that's ready to use at the edge.To do this we had to make sure the edge binary was really goodWe went through a lot of steps to make this happen.
We followed a process to optimize our research model for use at the edge:our research model had to be good enough, for the edge and that is what we focused on.I worked on Post-Training Quantization, which is also known as Post-Training Quantization. So I took the model. I changed the model from FP32, to INT8 precision. This was a thing because it made the model smaller. The Post-Training Quantization really helped the model. It went from 943 KB to 627.7 KB, which's a big difference. The Post-Training Quantization made the model use space.Deployment: Ported the model using the NXP eIQ toolkit, generating the instruction stacks necessary for execution on NXP crossover MCUs.

- Hardware Validation

The model gets it most of the time it is correct 87.57 percent of the time. We checked this with the hardware to make sure. The model is also really small; it is 627.7 KB. This is good because it means the model can fit inside the memory of the target hardware.

The model needs to work fast; it needs to be able to do things in less than one millisecond. This is important for lines where the model needs to work very quickly. The model is able to do this. The model meets the

requirements for speed lines, where things need to happen very quickly and the model is able to keep up with this.

Innovation and Uniqueness

- Engineering Resilience

Our project is really something because it can deal with things like a war story. The project can handle situations like a war story. That is what makes the project so interesting. The project is really special because it can take care of things, like a war story.

We were working on our project. We had a problem with some libraries. The libraries were giving us trouble when we were trying to get our project done. Our project was not going well because of the libraries.

The libraries were not working in the eiq environment. I had some problems with the libraries, in the eiq environment. The eiq environment was having trouble with the libraries.

We could have gone back to a version of these libraries. It would have been nice to do that with the libraries. Going back to a version of the libraries was something we could have done.

We did not want to wait for someone to do it. So we decided to fix the problem ourselves. This way the problem will get fixed faster. We can fix the problem ourselves. That is what we did.

We made some changes to the site-packages so that we can fix this problem with the site-packages.

This really helped to make sure that our validation pipeline for our project was stable. Our validation pipeline for our project is now working well because of this. Our project is going to be okay because our validation pipeline for our project is stable.

Our project has a validation pipeline that needs to be stable. This fix was really helpful for our project. We understand what we are doing when it comes to engineering. The validation pipeline is something that we're very good at. We also know a lot about the eiq environment and the global site-packages. This is because we have a lot of skill and knowledge in these areas.

We are very familiar with the validation pipeline and the eiq environment and the global site-packages. Our project and the validation pipeline are things that we work with all the time. The eiq environment and the global site-packages are also things that we work with regularly. We are really good at solving problems. We use the validation pipeline and the eiq environment to help us. We are also using the site-packages. This is how we get things done with the validation pipeline and the eiq environment and the global site-packages.

- Deterministic Performance

By ensuring TCM residency, we offer a competitive advantage over standard Edge-AI implementations. Our solution provides zero-wait-state performance, which is vital for the deterministic timing required by synchronous industrial control loops in semiconductor fabrication.