



B E E T L E B O X

BeetleboxCI: The new automation
platform for FPGA design



The speed of AI innovation is outpacing silicon chips. FPGA Accelerators are the solutions.

Here at Beetlebox, we envision a future where reconfigurable hardware is used to optimise not just the AI but provide whole application acceleration.

Cutting-edge whole application acceleration relies on constant improvement.

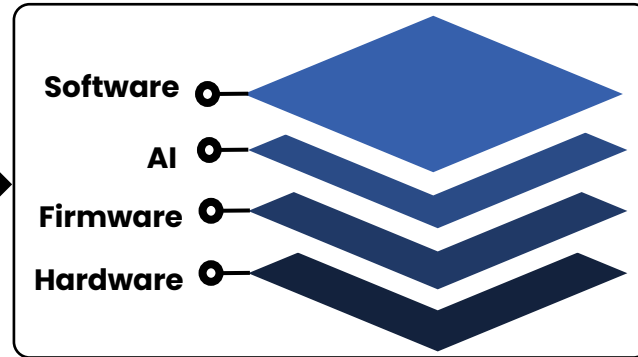
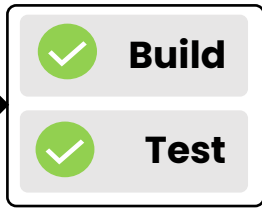
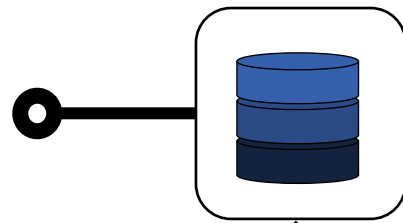
To keep up with AI innovation accelerated systems must have short development cycles, constant updates and increased testing and dependability.

Constant improvement requires automation.

To keep constantly improving, we need a system that can automate the entire development process from individual's developers code to final production.

Version Control

The system needs to integrate with standard version control systems such as Git.

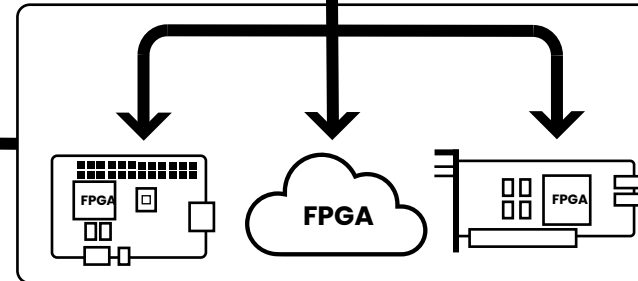
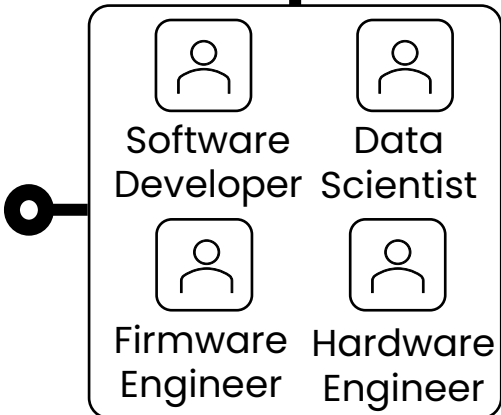


Full Technology Stack

The system must ensure the entire technology stack functions perfectly together.

Support for diverse teams

Every tool used by the team must be setup easily and run efficiently.



Deployment

The system must deploy to any cloud, on-premise or edge FPGA device.

Automating should be simple, scalable and cost-efficient.



Simple

The development team needs to be focused on their sections, not on trying to manage their infrastructure.

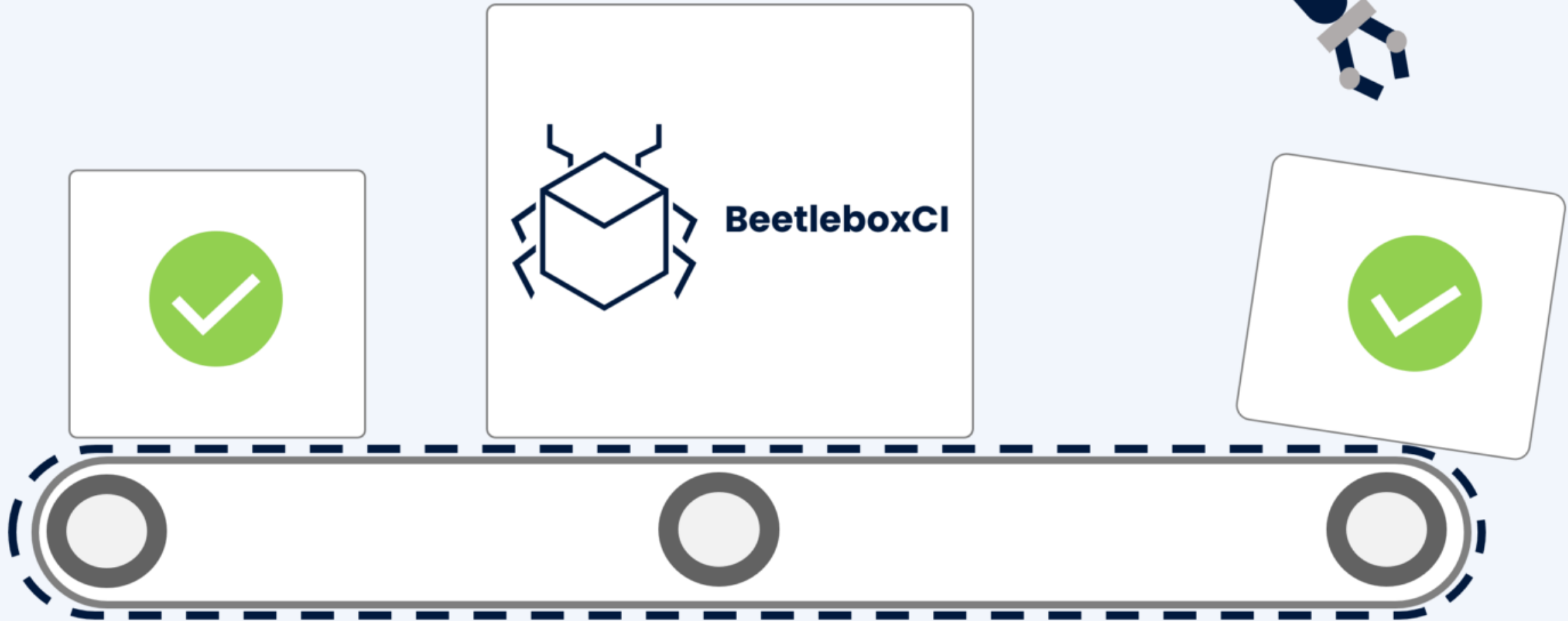
Scalable

Running across an entire technology stack requires hundreds of tests that have different computational demands. Servers need to scale with this demand.

Cost Efficient

The solution should not need large investment in infrastructure nor spending months configuring a cloud to a team's needs.

The Solution.



Design, develop and deploy AI with the automation platform for FPGA accelerator design

BeetleboxCI provides a solution that meets engineers' needs.



Simplicity

Native support for FPGA development tools with no need for any infrastructure setup.

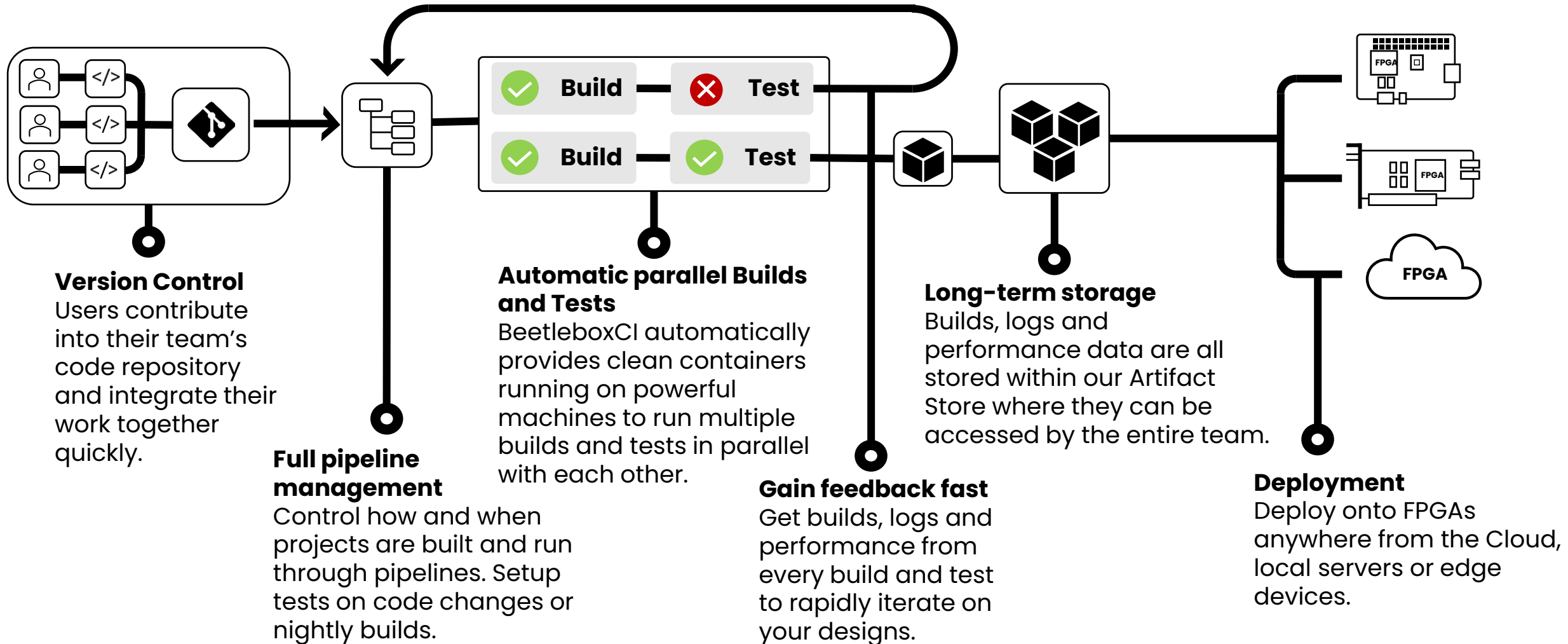
Scalability

BeetleboxCI is designed to handle all stages of development from one or two prototype tests to weeklong simulations.

Cost-efficiency

Developers are only charged for what they use and do not need to invest in costly infrastructure.

How BeetleboxCI Works.



Version Control
Users contribute into their team's code repository and integrate their work together quickly.

Full pipeline management
Control how and when projects are built and run through pipelines. Setup tests on code changes or nightly builds.

Automatic parallel Builds and Tests
BeetleboxCI automatically provides clean containers running on powerful machines to run multiple builds and tests in parallel with each other.

Gain feedback fast
Get builds, logs and performance from every build and test to rapidly iterate on your designs.

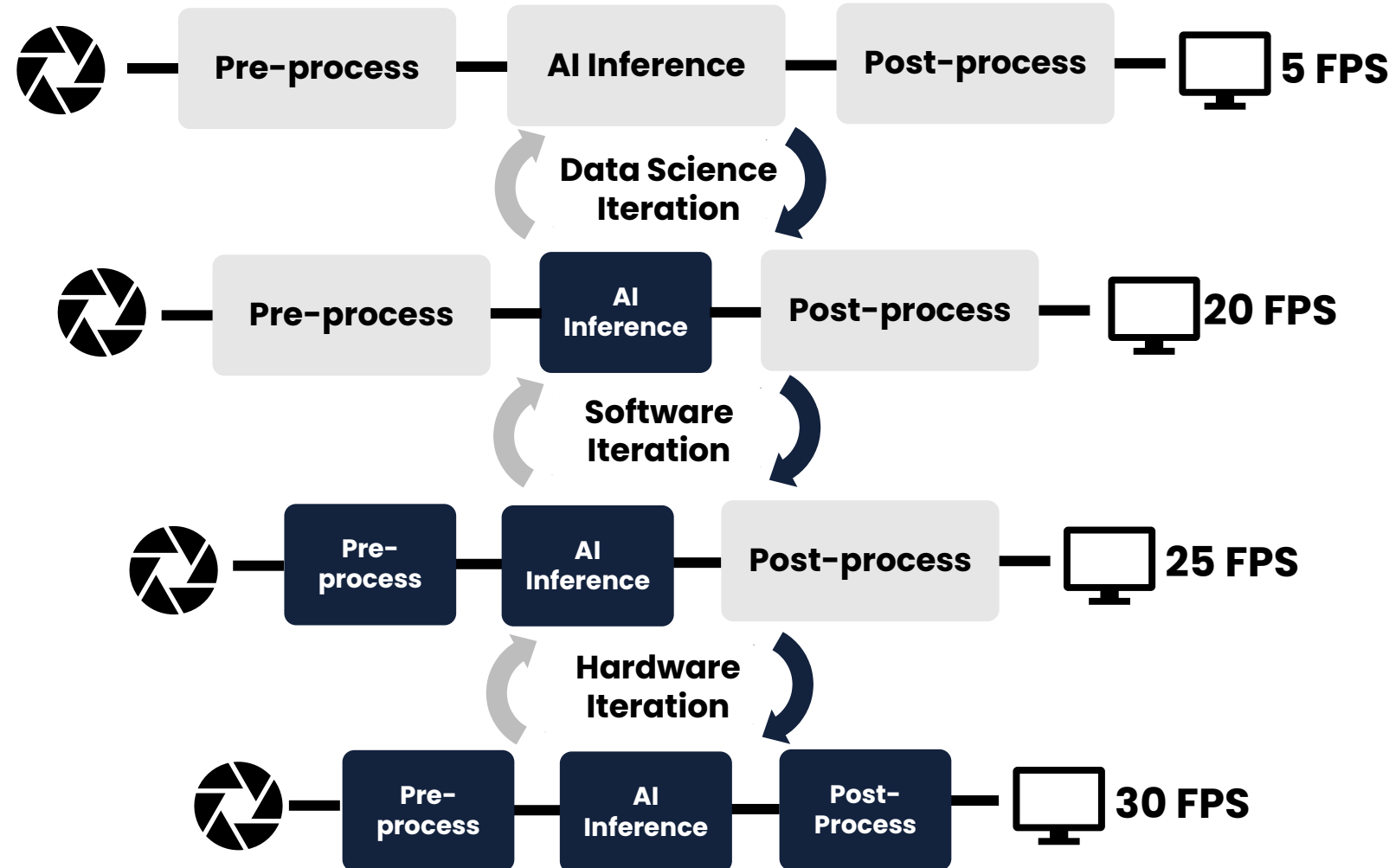
Long-term storage
Builds, logs and performance data are all stored within our Artifact Store where they can be accessed by the entire team.

Deployment
Deploy onto FPGAs anywhere from the Cloud, local servers or edge devices.

The impact of BeetleboxCI on AI.

Support at every level

BeetleboxCI supports an iterative process that allows developers to accelerate each section of the system within a managed environment, whilst also building and testing the entire system.

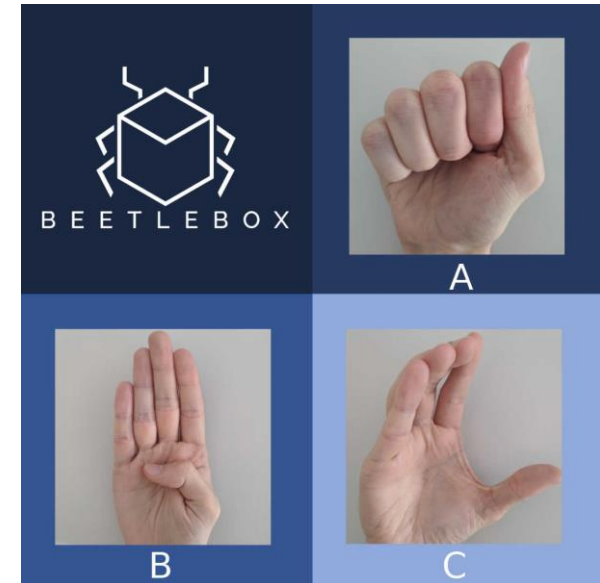


Example Design: Sign Language MNIST.

Challenge

Based off the original MNIST dataset, the sign language MNIST dataset contains 27,455 examples of 24 letters (excluding J and Z).

The challenge lies in correctly identifying these letters in a cost efficient manner to provide pragmatic help to the deaf and hard of hearing.

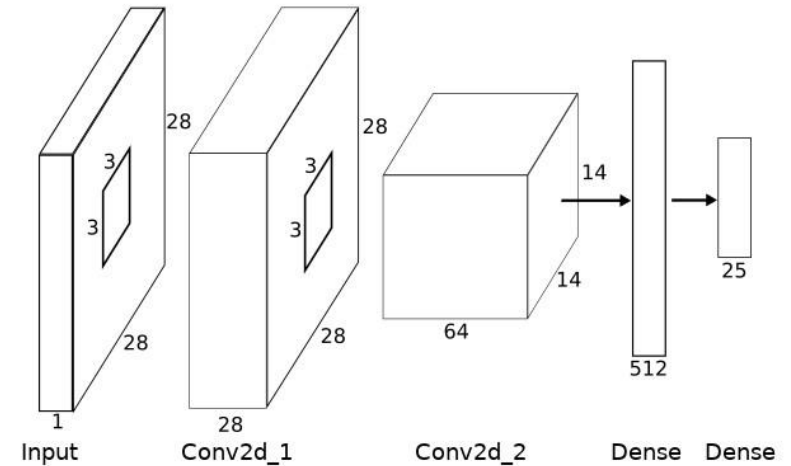


Example Design: Sign Language MNIST.

Solution

Using BeetleboxCI, we designed a simplistic neural network that can easily be trained on a CPU.

The design of neural networks is an iterative process, based on making improvements to the code and dataset itself. Our automated pipeline would retrain the neural network, quantize and prepare the files on FPGAs within three minutes of submitting a change to the code repository.



Example Design: Sign Language MNIST.

Result

On a ZCU104:

- *Throughput : 1067.21 FPS*
- *Accuracy : 92.95%*

Entire project is available as open source and can be setup in as little as twelve steps:

https://github.com/beetleboxorg/sign_language_mnist

Get Developing Now



Get started for free on our website with 10,000 free credits: beetlebox.org

Reach us at:

02039816275

contact@beetlebox.org

Unit 1. 10,
Chester House,
Kennington Park,
1-3 Brixton Road,
London,
SW9 6DE