brainchip
event-based computing

Akida Edge AI Box

Model Deployment Guide

Version 1.0

1. Introduction

Welcome to the Akida Edge AI Box Model Deployment Guide. This manual will walk you through the process of deploying custom machine learning models to your Akida Edge AI Box. Whether you're using pre-trained models from MetaTF or developing your own with Edge Impulse, this guide will help you get your models up and running.

2. What You'll Need

- Akida Edge AI Box (set up and functioning per the Akida Edge AI Box User Manual)
- Host computer connected to the same network as the Akida Edge AI Box
- Secure file transfer application (e.g., WinSCP) installed on your host computer
- Custom model file (.bin format) from MetaTF or Edge Impulse

3. Model Development Platforms

The Akida Edge AI Box supports models developed on two primary platforms:

BrainChip MetaTF

MetaTF (https://brainchip.com/support, select “Meta TF Documentation”) is a comprehensive machine learning framework developed by BrainChip specifically for creating, training, and testing neural networks for the Akida platform. It provides Python libraries and examples that are consistent with and leverage the TensorFlow framework.

Edge Impulse

Edge Impulse (https://docs.edgeimpulse.com/docs/edge-ai-hardware/cpu-+-ai-accelerators/akd1000) is a leading development platform for machine learning on edge devices. It offers a complete, cloud-based solution for training models specifically optimized for edge deployment, including support for Akida devices.

4. Pre-installed Models

Your Akida Edge AI Box comes with five pre-installed models:

- Face Detection (FOMO)
- Vitamin Classification (FOMO)
- Person/Car Detection (YOLOv2)
- Face Detection (YOLOv2)
- Visual Wake Word (AkidaNet)

These models are located in the `/etc/akida` directory on your Edge AI Box.

5. Model Specifications

Face Detection with FOMO (Faster Objects, More Objects)

Description: FOMO is an object detection algorithm developed by Edge Impulse, optimized for constrained devices. This model is trained to detect faces.

File Name: `/etc/akida/fomo_face_model.bin`

Input: Image (160x160 pixels)

Output: List of centroid coordinates representing face locations

Vitamin Classification with FOMO

Description: This model is trained to detect and classify different types of vitamins.

File Name: `/etc/akida/fomo_model.bin`

Input: Image (160x160 pixels)

Output: List of centroid coordinates for four vitamin classes: "K2", "Vitamin B", "Vitamin C", and "Zinc"

Person/Car Detection with YOLOv2

Description: This model uses the YOLO (You Only Look Once) architecture to detect people and cars in images.

File Name: `/etc/akida/yolov2_model.bin`

Input: Image (224x224 pixels)

Output: Bounding boxes and class labels for "person" and "car"

Face Detection with YOLOv2

Description: This model uses the YOLO architecture specifically for face detection.

File Name: `/etc/akida/yolo_face_model.bin`

Input: Image (224x224 pixels)

Output: Bounding boxes for the "face" class

Visual Wake Word with AkidaNet

Description: This model uses the AkidaNet architecture to detect the presence of a person in an image.

File Name: `/etc/akida/vww_model.bin`

Input: Image (96x96 pixels)

Output: Binary classification: "person" or "no person"

6. Deploying a New Model

When deploying a new model, the input size and model classes need to be identical to the respective models above.

Prepare Your Model

Ensure your custom model is in the correct .bin format:

MetaTF model file preparation

Use the following Python script to convert your model to a .bin file:

```
#!/usr/bin/env python
import os
from akida import AKD1000
# Assuming variable 'model' is a pre-trained Akida model.
# Map/compile converted model for a two node IP
model.map(device=AKD1000())
# Confirm model mapping: NP allocation and binary size
model.summary()
# Retrieve model program binary
program = model.sequences[0].program
# Generate a binary file for the model
with open('model.bin', 'wb') as file:
    file.write(program)
```

Edge Impulse model file preparation

Download the Akida-compatible .bin file from the deployment page. For detailed instructions, refer to the Edge Impulse Akida Deployment Guide:

https://docs.edgeimpulse.com/docs/edge-ai-hardware/cpu-+-ai-accelerators/akd1000#brainchip-metatf-deployment-block

Transfer the Model

1. On the laptop, open your secure file transfer application (e.g., WinSCP).
2. Connect to your Akida Edge AI Box using its IP address.
3. Navigate to the `/etc/akida` directory on the Edge AI Box.
4. Upload your .bin file to this directory.

Update Model Configurations

Configuration settings for the Akida Edge AI Box demonstrations are managed from a configuration file. The following instructions walk through the process of locating the file to update (if desired).

1) Connect to your Akida Edge AI Box via SSH:

```
ssh root@<Akida_Edge_AI_Box_IP>
```

2) Navigate to the Akida configuration directory:

```
cd /etc/akida
```

3) Open the configuration file:

```
nano akidademo.conf
```

4) Update the relevant model path and name. For example:

```
"akida1_model_path": "/etc/akida/your_new_model.bin",
"akida1_model_name": "your_model_name",
```

5) Save and exit the file (Ctrl+X, then Y, then Enter).
6) Reboot the Akida Edge AI Box for changes to take effect.

7. Troubleshooting

If you encounter issues:

- Double-check that your .bin file is in the correct format for Akida.
- Ensure the file path in the configuration file matches your new model's filename.
- Verify that the model name in the configuration file is correct.
- Check the Edge AI Box's system logs for any error messages.

For further assistance, please consult the full Akida Edge AI Box User Manual or contact BrainChip support.