



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica e Informática

Relatório de Estágio
**Laboratório de Interface Homem Máquina - Universidade
Federal de Campina Grande**

Arthur de Lima Carneiro

Campina Grande, PB
Maio de 2021

Arthur de Lima Carneiro

**Laboratório de Interface Homem Máquina - Universidade
Federal de Campina Grande**

*Relatório de Estágio Supervisionado
submetido à Unidade Acadêmica de
Engenharia Elétrica da Universidade
Federal de Campina Grande como
parte dos requisitos necessários para
a obtenção do grau de Bacharel em
Engenharia Elétrica*

Área de Concentração: Controle e Automação

Orientador: Danilo Freire de Souza Santos

Campina Grande, PB
Maio de 2021

Arthur de Lima Carneiro

**Laboratório de Interface Homem Máquina - Universidade
Federal de Campina Grande**

*Relatório de Estágio Supervisionado
submetido à Unidade Acadêmica de
Engenharia Elétrica da Universidade
Federal de Campina Grande como
parte dos requisitos necessários para
a obtenção do grau de Bacharel em
Engenharia Elétrica.*

Área de Concentração: Controle e Automação

Aprovado em ___/___/___

Jaidilson Jó da Silva

Universidade Federal de Campina Grande
Avaliador

Danilo Freire de Souza Santos

Universidade Federal de Campina Grande
Orientador

*Sempre que te perguntarem se podes fazer um trabalho,
respondas que sim e te ponhas em seguida a aprender como se faz.*

Agradecimentos

Gostaria de agradecer a todos os membros do projeto de capacitação Huawei do Brasil: Mohit, Igor, Ricardo, Toni e Allan pelo tratamento profissional e pelas intervenções necessárias para que este trabalho pudesse ter resultado satisfatório. Agradecer ainda ao meu professor orientador Danilo Freire de Souza Santos que apresentou a atividade a mim e confiou em minha capacidade de tentar resolvê-la. Aprendi bastante tecnicamente e pessoalmente nesse período de estágio com a colaboração de vocês.

Resumo

Esse trabalho apresenta as atividades desenvolvidas na disciplina de Estágio Supervisionado pelo discente Arthur de Lima Carneiro no Laboratório de Interface Homem Máquina (LIHM) na Universidade Federal de Campina Grande (UFCG). O estágio teve por objetivo a instalação de um laboratório remoto voltado para inteligência artificial utilizando o *hardware* Atlas 200 *Development Kit* desenvolvido pela Huawei e elaboração de um material didático auxiliar.

Palavras chave: Inteligência Artificial, Ensino Remoto, Computação em Borda, Atlas 200 DK.

Abstract

This report presents the activities developed in the supervised internship of the Electrical Engineering student Arthur de Lima Carneiro, held at the Human-Machine Interface Laboratory (LIHM) at Federal University of Campina Grande (UFCG). As a internship activity the student had to install a remote laboratory focused on Artificial Intelligence using the Atlas 200 Development Kit made by Huawei and create auxiliary teaching material.

Keywords: Artificial Intelligence, Remote Teaching, Edge Computing, Atlas 200 DK.

Lista de Figuras

2.1	Fotografia do salão principal do LIHM.	14
3.1	Vista frontal do Atlas 200 DK.	16
3.2	Vista em perspectiva do Atlas 200 DK	16
3.3	Representação da arquitetura do sistema do Atlas 200 DK.	17
3.4	Imagem da placa do Atlas 200 DK com destaque para os 4 LEDs.	18
3.5	Fluxograma da conversão de modelos com módulo ATC.	19
3.6	Representação da arquitetura do Mindstudio.	20
4.1	Representação da arquitetura de Montagem de cada estação do laboratório.	21
4.2	Log de execução do projeto com identificação das imagens.	22
4.3	Fotografia da estrutura do Laboratório com 2 estações de trabalho utilizado no projeto piloto.	24
4.4	Fotografia de atividade com aluno atuando remotamente no laboratório.	24
4.5	Fotografia de atividade com aluno e instrutor atuando remotamente no laboratório.	25
6.1	Arquivos necessários para instalação do Atlas 200 DK.	32
6.2	Saídas para os comandos do terminal	35
6.3	Janela de inicial do MindStudio.	35
6.4	Modo alternativo para abertura de novo projeto no MindStudio.	36
6.5	Tela de abertura de novo projeto no Mindstudio.	36
6.6	Tela de para selecionar banco de dados pré-existente.	37
6.7	Criação de nova pasta no projeto.	37
6.8	Transferência do Arquivo resnet50 para a pasta caffe_model.	38
6.9	Opções de conversão de modelo de Caffé para Mindspore.	39
6.10	Opção para adicionar modelo Mindspore ao projeto.	39
6.11	Localização do modelo convertido para adicionar ao projeto.	40
6.12	Localização do modelo convertido para adicionar ao projeto.	41
6.13	Log de execução do projeto com identificação das imagens.	42

Lista de Tabelas

- 1 Portfólio dos modelos Atlas da Huawei e os processadores de cada um. 15

Lista de Abreviaturas e Siglas

- ARM Advanced RISC Machine
- ANEEL Agência Nacional de Energia Elétrica
- AIPP Artificial Intelligence Preprocessing
- ATC Ascend Tensor Compiler
- CANN Compute Architecture for Neural Networks
- DK Developer Kit
- DVPP Digital Vision Preprocessing
- GE Graph Engine
- IDE Integrated development environment
- IA Inteligência Artificial
- I2C Inter-Integrated Circuit
- LIHM Laboratório de Interfaces Homem-Máquina
- RGB Red-Green-Blue
- RNA Redes Neurais Artificiais
- SPI Serial Peripheral Interface
- UART Universal Asynchronous Receiver-Transmitter
- UFMG Universidade Federal de Campina Grande

Sumário

1	Introdução	13
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	13
2	Laboratório de Interface Homem-Máquina - LIHM	14
3	Revisão Tecnológica	15
3.1	Atlas 200 DK	15
3.2	MindStudio	18
4	Atividades Realizadas	20
4.1	Estudo dos Equipamentos	20
4.2	Projeto de Laboratório Remoto	21
4.3	Projeto Piloto	23
5	Conclusão	26
6	Apêndice	28
6.1	Instalação do Atlas 200 DK e Mindstudio	28
6.1.1	Material Necessário	28
6.1.2	Procedimentos	28
6.2	Experimento 1 - Conversão de modelos Caffe com Mindstudio	34
6.2.1	Apresentação da Atividade	34
6.2.2	Materiais	34
6.2.3	Procedimentos	34
6.2.4	Discussões	42
6.3	Experimento 2 - Acesso via SSH e Execução por linha de comando	44
6.3.1	Apresentação da Atividade	44
6.3.2	Materiais	44
6.3.3	Procedimentos	44
6.3.4	Discussões	45

1 Introdução

O presente relatório descreve as atividades que foram desempenhadas na disciplina de Estágio Supervisionado Obrigatório para a obtenção do título de Bacharel em Engenharia Elétrica pela UFCG. A duração foi de 360 horas e foi realizado no Laboratório de Interface Homem Máquina (LIHM) - UFCG durante o período de 04 de Fevereiro de 2021 a 04 de Maio de 2021 sob orientação do professor Danilo Freire de Souza Santos e supervisão do professor Gutemberg Gonçalves dos Santos Junior.

O estágio curricular obrigatório presente na grade curricular do curso de graduação em Engenharia Elétrica busca que o discente tenha uma experiência prática dos tópicos explorados em salas de aula. Além disso, é possível aprofundar-se um pouco mais em assuntos que não tenham sido abordados nas disciplinas. A partir disso, este trabalho apresenta as atividades desenvolvidas no estágio, quais foram as habilidades exigidas no mesmo.

A motivação desse estágio foi o crescente interesse dos alunos de graduação em tópicos relacionados a Inteligência Artificial (IA), mais especificamente a área de *machine learning*. Essa área tem crescido bastante em diversos segmentos e vive em constante integração com a sociedade como um todo. Os *chatbots* de aplicativos bancários e assistentes pessoais (Alexa e Siri) são exemplos disso, mas há muitas outras aplicações dos mais variados tipos.

Dessa forma, o laboratório *Embedded* elaborou um curso introdutório como capacitação para uma prova de certificação da *Huawei Technologies* em IA abordando tópicos elementares de *machine learning*, principais frameworks utilizados e apresentando os produtos da empresa que são utilizados em soluções de problemas que envolvem IA.

Apesar de apresentar os produtos e os possíveis problemas que eram capazes de resolver, faltava uma abordagem em uma perspectiva prática para os alunos que realizavam o curso. Portanto, foi proposto ao estagiário a montagem de um laboratório para realização de uma atividade prática com os alunos utilizando os equipamentos *Huawei* adquiridos pelo *Embedded*.

O fator desafiador foi a modalidade do laboratório. Em virtude da pandemia de COVID-19, todas as atividades deveriam ser realizadas remotamente. Isso quer dizer que os equipamentos deveriam ser instalados, acessáveis de qualquer lugar e com a possibilidade de realizar as atividades propostas.

1.1 Objetivo Geral

Planejamento e instalação de um laboratório explorando as funcionalidades das soluções *Huawei* e o Atlas 200 DK como ferramenta didática de ensino.

1.2 Objetivos Específicos

- Elaborar inventário dos materiais necessários ao laboratório;
- Pesquisar soluções para instalação dos equipamentos;
- Pesquisar possíveis experimentos a serem realizados;
- Pesquisar ferramentas que permitam acesso remoto seguro aos equipamentos;
- Elaborar material didático que permita reprodução do laboratório.

2 Laboratório de Interface Homem-Máquina - LIHM

O Laboratório de Interface Homem-Máquina é um ambiente em que se realiza pesquisa e desenvolvimento de soluções nos níveis de graduação e pós-graduação. O laboratório dispõe de um salão principal de $70 m^2$ (Figura 2.1) e um ambiente de testes. No primeiro, dispõe-se de diversas estações de trabalho com um computador e o ambiente de testes. As atividades são focadas no desenvolvimento de soluções em software e hardwares voltadas para a interface de usuário visando ambientes de automação industrial. O objetivo principal é reduzir a possibilidade de erro humano na operação de sistemas e produtos.

Além da pesquisa o LIHM também tem fins de ensino com a realização de atividades das disciplinas Informática Industrial na graduação em Engenharia Elétrica e Interfaces Homem-Máquina, Avaliação da Usabilidade de Produtos e Informática Industrial na pós-graduação em Engenharia Elétrica.

Em virtude da pandemia de COVID-19 algumas adaptações foram necessárias em razão da biossegurança. Este trabalho foi realizado em uma sala menor em que o estagiário trabalhava sozinho e recebia instruções remotamente da equipe de trabalho.



Figura 2.1: Fotografia do salão principal do LIHM.

3 Revisão Tecnológica

Esse capítulo abordará os principais equipamentos utilizados para as atividades do estágio. Sendo o *hardware* Atlas 200 DK e o *software* MindStudio.

3.1 Atlas 200 DK

A plataforma Atlas é uma parte da estratégia de produtos da Huawei para Inteligência Artificial. A proposta da empresa é possuir um modelo *fullstack*, ou seja, dispor de produtos em todas as etapas de desenvolvimento. Desse modo, a Huawei dispõe de um portfólio vasto de produtos desde chips dedicados para IA até produtos com aplicações específicas. O principal objetivo com esses modelos é a aproximação do alto desempenho computacional na borda [2].

A Huawei dispõe de uma família de produtos dividida em 2 grupos. Os produtos que possuem o processador Ascend 310 que é voltado para inferência de modelos pré-treinados e os que possuem o processador Ascend 910 que é voltado somente para o treinamento de modelos. Ambos os processadores são da família de arquiteturas ARM. A Tabela 1 mostra os produtos da plataforma Atlas e quais os processadores que cada um tem integrado em sua arquitetura.

Produto	Processador
Atlas 200 DK - Model:3000 Atlas 200 AI - Model:3000 Atlas 300 AI Accelerator Card - Model:3000 Atlas 500 AI Edge Station - Model:3000 Atlas 800 AI Server - Model:3000/3010	Ascend 310
Atlas 300 AI Accelerator Card - Model:9000 Atlas 800 AI Server - Model:9000/9010 Atlas 900 AI Cluster	Ascend910

Tabela 1: Portfólio dos modelos Atlas da Huawei e os processadores de cada um.

Nesse trabalho foi utilizado o Atlas 200 DK, que possui o processador Ascend 310 em sua arquitetura. O produto utiliza-se do mesmo *core* do Atlas 200 AI com a vantagem de possibilitar maior flexibilidade aos desenvolvedores para aplicações. Pode ser utilizado para pesquisa acadêmica, para fins educacionais e como protótipo de soluções [2].

Abaixo temos algumas das principais especificações desse produto:

- Design compacto, leve e de fácil uso;
- Baixo consumo de energia;
- Porta de Rede RJ-45;
- Porta USB tipo C;
- Entrada para cartão micro-SD, compatível com SD 3.0;
- Conector GPIO de 40 entradas e saídas (E/S) para comunicação.

Alguns possíveis cenários de aplicação do Atlas 200 DK são as seguintes:

- Integrável com *Huawei Cloud* para o treinamento e deploy de aplicações;
- Ideal para análises de grandes conjuntos de dados de fotos ou vídeos;
- Aplicável em soluções de vigilância e integrável com drones e robôs.

Nas Figuras 3.1 e 3.2 temos imagens do Atlas 200 DK frontal e em perspectiva, respectivamente. Detalhe para a Figura 3.2 em que observa-se da esquerda para a direita as seguintes entradas: entrada para fonte de energia, porta USB-C, entrada para cartão micro-SD e porta de rede RJ-45.



Figura 3.1: Vista frontal do Atlas 200 DK.
[4]



Figura 3.2: Vista em perspectiva do Atlas 200 DK
[2]

A arquitetura do sistema do Atlas 200 DK está mostrada na Figura 3.3 com os respectivos protocolos de comunicação internos para conectar cada um dos módulos. O Atlas 200 é o módulo

principal que conecta todas as entradas do sistema. O módulo Hi3559C é dedicado para processamento dados multimídia (áudio e vídeo). Aliado às entradas de câmera por cabo flat e os microfones nativos torna-se uma ferramenta poderosa para o Atlas 200 DK. Apesar da especialidade em análise de dados de áudio e vídeo, a placa de 40 pinos de E/S permite a conexão de diversos sensores através de diversos protocolos de comunicação como I2C, UART e SPI.

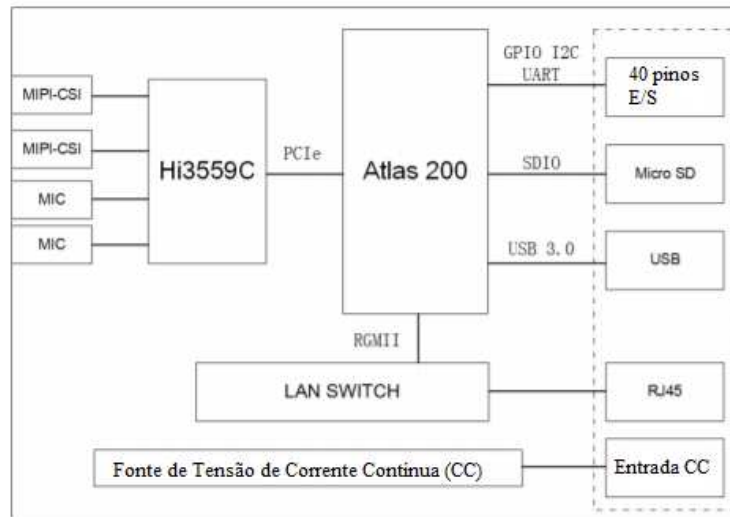


Figura 3.3: Representação da arquitetura do sistema do Atlas 200 DK.
[3]

Na Figura 3.4 é mostrado o Atlas retirado a sua tampa para ter acesso às entradas seriais. Em evidência estão os 4 LEDs, que são importantes para saber o status atual do equipamento.

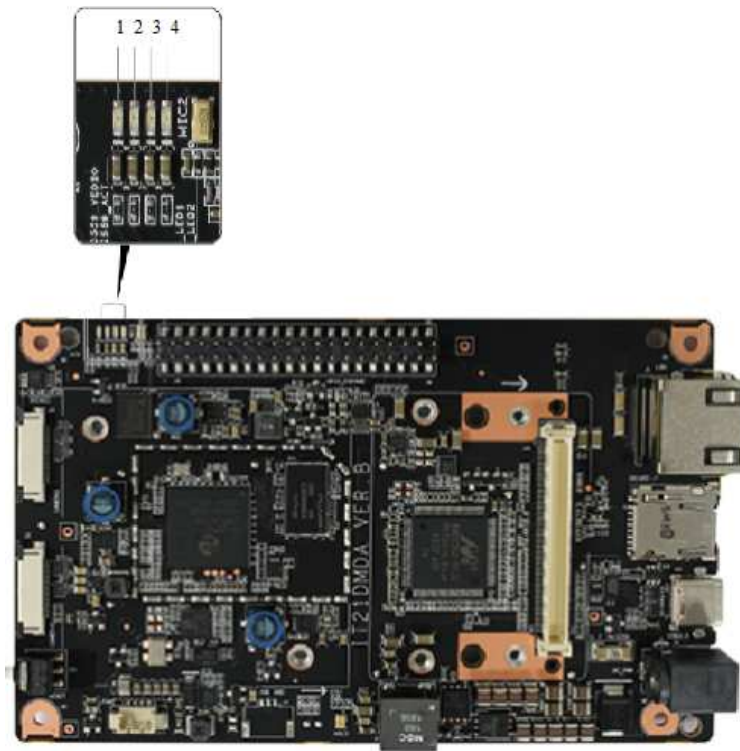


Figura 3.4: Imagem da placa do Atlas 200 DK com destaque para os 4 LEDs.
[2]

3.2 MindStudio

O Mindstudio é um software de desenvolvimento no formato End-to-End (E2E) voltado para as soluções Huawei de IA. Em outras palavras, essa plataforma permite desenvolvimento, debug e otimização de aplicações [3]. Atualmente encontra-se na versão 2.0.0 e é compatível somente com o sistema operacional Ubuntu 18.04. Entretanto, sua documentação está em constante atualização e espera-se em breve uma compatibilidade mais abrangente.

O software permite uma avaliação vasta de modelos de IA, sobretudo Redes Neurais Artificiais (RNA). A funcionalidade Ascend Tensor Compiler (ATC) permite que modelos desenvolvidos em diversos frameworks sejam convertidos para o Mindspore (framework compatível com os processadores Ascend). Desse modo, é permitido que ainda que um desenvolvedor não saiba utilizar as bibliotecas do Mindspore para construir um modelo é possível que utilize os produtos Atlas. Na Figura 3.5 é visto o fluxograma de funcionamento da conversão do modelo.

Essa funcionalidade e versatilidade do Mindspore através do Mindstudio será um dos pontos explorados nas atividades desse trabalho.

Pode-se afirmar que o Mindstudio é um HUB de desenvolvimento pelas funcionalidades já citadas. Além disso, uma particularidade é que a execução dos projetos e aplicativos do software são feitas em um dispositivo Atlas que encontra-se conectado via Ethernet ou USB.

O Mindstudio destaca-se também por outras funcionalidades bastante úteis ao desenvolvedor. Como as seguintes:

- No Modo de Visualização de modelos é possível fazer a leitura de redes neurais de diversos frameworks distintos, inclusive com capacidade de edição com adição e exclusão de camadas.
- Artificial Intelligence Preprocessing (AIPP) é um módulo responsável por auxiliar o módulo DVPP. É responsável por realizar diversas operações com as imagens e ainda converter a saída do módulo DVPP em imagens no formato RGB.
- Ascend Computing Learning (ACL) é uma linguagem que utiliza funções de C e C++ para aplicações em modelos de deep learning utilizando de programação orientada a objetos.
- Graph Engine (GE) é um módulo nativo dos softwares e dispositivos Huawei que é facilita a criação de estruturas em formas de grafos. Desse modo que as Redes Neurais Artificiais são mostradas no MindStudio.

A arquitetura do software do Mindstudio é mostrada na Figura 3.6

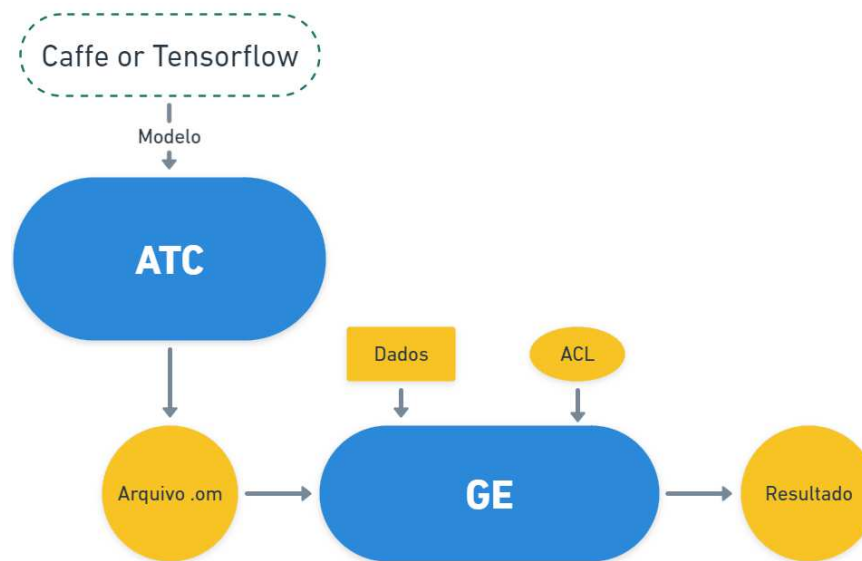


Figura 3.5: Fluxograma da conversão de modelos com módulo ATC.

[3]

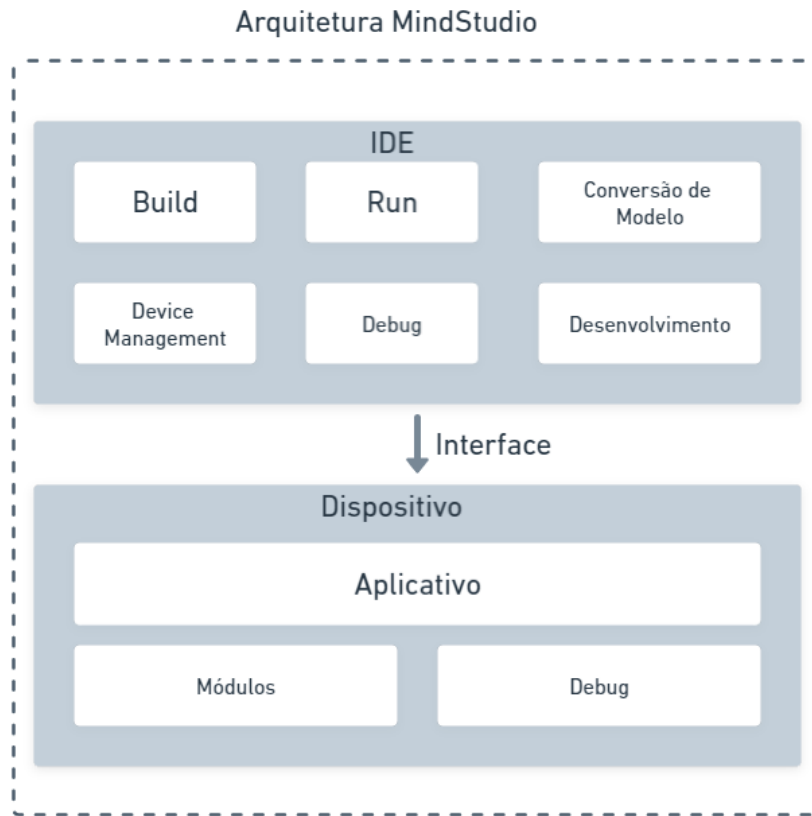


Figura 3.6: Representação da arquitetura do Mindstudio.
[3]

4 Atividades Realizadas

Durante o estágio foi proposto um plano de trabalho dividido em diversas etapas para que se obtivesse êxito ao fim da atividade. Inicialmente foi apresentada a equipe de trabalho, o objetivo final do projeto e os prazos necessários de cada etapa. Foi apresentado o cronograma, os equipamentos a serem utilizados, o público alvo e o período em que um piloto deveria ser realizado.

4.1 Estudo dos Equipamentos

O estágio iniciou-se com a falta dos equipamentos e exigiu um estudo preliminar de documentação. Esse período foi bastante proveitoso, pois no momento em que os produtos chegaram ao laboratório já havia familiaridade com as funcionalidades. A comunidade de desenvolvimento e a fabricante *Huawei* tem atualizado semanalmente alguns materiais e descrição de projetos em seus canais oficiais de fóruns e repositórios.

Desse modo mesmo que não se tenha acesso aos Atlas 200 DK e MindStudio é possível estudar a respeito das funções. Há alguns projetos da comunidade de buscar virtualizar alguns modelos da plataforma Atlas para que possam ser acessados via Huawei Cloud, como já acontece com a Raspberry Pi na plataforma Azure.

A ideia desta etapa foi buscar junto a documentação oficial e debates da comunidade de desenvolvimento uma trilha de configuração que levasse ao mínimo de bugs possível. Durante a execução desta etapa do projeto o suporte da fabricante incorporou um guia de um desenvolvedor à documentação oficial em que algumas variáveis de ambiente eram consertadas de modo a evitar problemas futuros com o funcionamento do Atlas 200 DK e MindStudio.

Nessa etapa foi encontrado também que a nova versão do Framework Tensorflow 2.0 não está compatível no MindStudio 2.0.0 beta2 para a conversão dos modelos para o framework nativo do Atlas, o Mindspore. Desse modo, houve uma limitação de projeto a respeito da conversão dos modelos, pois a comunidade que utiliza Tensorflow não utiliza mais tanto as versões 1.X.

4.2 Projeto de Laboratório Remoto

Um dos fatores limitantes causados pela pandemia de COVID-19 ao projeto foi a impossibilidade de funcionamento do laboratório com os alunos acessando diretamente os computadores com o Atlas 200 DK instalado.

Desse modo, foi proposto um modelo em que o estudante acessaria remotamente as máquinas e as utilizaria com o auxílio e inspeção direta do instrutor. O projeto foram disponibilizados os seguintes equipamentos para a instalação do laboratório:

- 1 Notebook Dell G5 com 16 GB de memória RAM e 512 GB SSD com Windows 10;
- 2 Notebook Avell A62 com 32 GB de memória RAM e 512 GB SSD com Windows 10;
- 5 Atlas 200 DK com fonte de energia;
- 5 cartões micro SD Sandisk de 16 GB;
- Cabos USB tipo C e Cabos Ethernet.

A partir da montagem de cada uma das máquinas e configuração das mesmas junto ao Atlas 200 DK segundo a arquitetura definida na Figura 4.1 foi elaborado um material auxiliar para a configuração das estações de trabalho do laboratório que pode ser encontrado nos apêndices deste trabalho na seção 6.1.

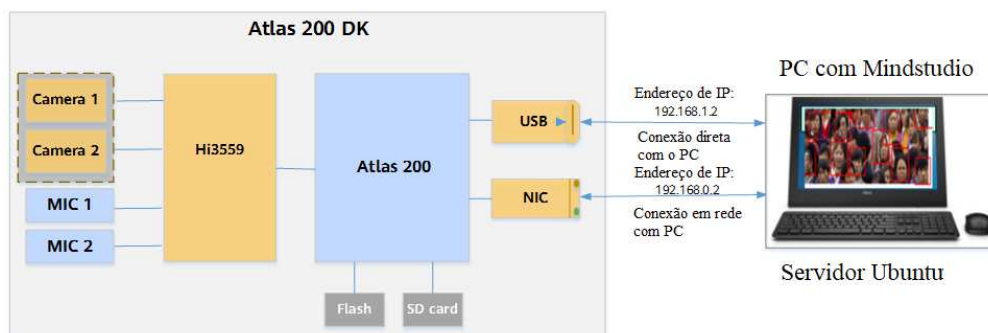


Figura 4.1: Representação da arquitetura de Montagem de cada estação do laboratório.

[4]

A listagem de material foi suficiente, pois equipe de trabalho conseguiu concluir que um computador poderia ser um Hub para mais de um Atlas 200 DK. Entretanto, para fins didáticos não será tão interessante porque não é possível 2 alunos utilizarem a mesma estação de trabalho para acessar janelas distintas do MindStudio para trabalhar em projetos.

Utilizou-se um experimento com parte dos códigos vindos nativamente nas pastas do MindStudio. O experimento utilizou-se do resnet-50, uma conhecida arquitetura de Rede Neural Artificial (RNA) da comunidade de desenvolvimento. Na documentação do Atlas foi possível encontrar diversos arquivos relativos a este projeto, como a arquitetura do modelo no framework caffe e o modelo treinado, também em caffe. A partir disso utiliza-se o MindStudio para converter o modelo do framework caffe para o framework Mindspore. Depois da conversão é enviado o modelo treinado e convertido, os dados de entrada e os scripts de saída para o Atlas 200 DK. Depois de algum tempo o Atlas 200 DK gera as respostas aos dados de entrada. Todo esse procedimento foi documentado para ser realizado com os alunos e encontra-se na seção 6.2.1. A versão documentada possui algumas modificações feitas pelo estagiário para fins didáticos. As classificações são explicitamente impressas na saída do projeto para que o aluno saiba se o modelo classificou corretamente sem necessitar consultar índices em um arquivo de 1000 classes, conforme mostrado na Figura 4.2.

```
2021-03-11 16:33:12 - [INFO] Assigning execute permission to run.sh on the remote
/home/HwHiAiUser/HIAI_PROJECTS/workspace_mind_studio/MyApp3_9f210d93/out/run.sh: 1
sudo: no tty present and no askpass program specified
[INFO] acl init success
[INFO] open device 0 success
[INFO] create context success
[INFO] create stream success
[INFO] get run mode success
[INFO] load model ../model/resnet50.om success
[INFO] create model description success
[INFO] create model output success
[INFO] start to process file:../data/dog1_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[161] - value[76.76%]
basset
[INFO] top 2: index[162] - value[15.48%]
beagle
[INFO] top 3: index[167] - value[3.85%]
English_foxhound
[INFO] top 4: index[163] - value[2.16%]
bloodhound
[INFO] top 5: index[166] - value[1.17%]
Walker_hound
[INFO] output data success
[INFO] start to process file:../data/dog2_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[267] - value[93.60%]
standard_poodle
[INFO] top 2: index[266] - value[4.11%]
```

Figura 4.2: Log de execução do projeto com identificação das imagens.

Posteriormente ao experimento relatado a equipe de trabalho concluiu que seria interessante explorar a funcionalidade de acessar o Atlas 200 DK via SSH e executar os projetos que estão no

mesmo via linha de comando. Apesar de ser um experimento relativamente simples em relação ao anterior é didaticamente importante para mostrar que o equipamento objeto de estudo pode ser utilizado em aplicações móveis como em drones ou um robôs, por exemplo.

4.3 Projeto Piloto

Para validação do laboratório foi feito um piloto com os alunos participantes da capacitação em machine learning para a prova de certificação da Huawei. O laboratório teve por objetivo complementar os conhecimentos adquiridos a partir de um contato prático com os conceitos e equipamentos estudados nas aulas teóricas. No total foram 29 participantes do projeto piloto em 23 encontros. Nele foram feitas pelos alunos as atividades propostas nos experimentos já citados.

A estrutura do laboratório foi possível ser utilizada remotamente pelo software TeamViewer, que consegue transmitir a tela, mouse e teclado para um usuário em qualquer lugar conectado pela internet. A escolha desse software se deu pela simplicidade de uso para todos os envolvidos. Era permitido, inclusive que o instrutor e o aluno acessassem remotamente o laboratório. Entretanto, o instrutor teria poderes de administrador da janela. Na Figura 4.3 é exibida a estrutura instalada no laboratório para as atividades propostas. Nas Figuras 4.5 e 4.4 estão documentados 2 encontros distintos, no primeiro o estagiário esteve no laboratório auxiliando o aluno que acessava remotamente e utilizando-se de ferramentas de conversa por áudio para se comunicar, já na segunda temos um encontro em que o instrutor e os alunos estão acessando remotamente as instalações do laboratório.

O resultado final foi satisfatório com todos os alunos conseguindo terminar as atividades no tempo proposto. Os dois experimentos foram realizados em 1 hora de atividades, independente do atendimento individual ou em duplas. Destaca-se que o senso crítico dos participantes foi importante para as discussões do experimento, pois foi proposto mais que uma vez que uma imagem diferente das previamente escolhidas fosse utilizadas para avaliar o modelo. Posteriormente a isso era possível que o aluno constatasse o funcionamento dos modelos desenvolvidos. Essa atividade foi posteriormente incluída no tópico de discussões do Experimento 1.

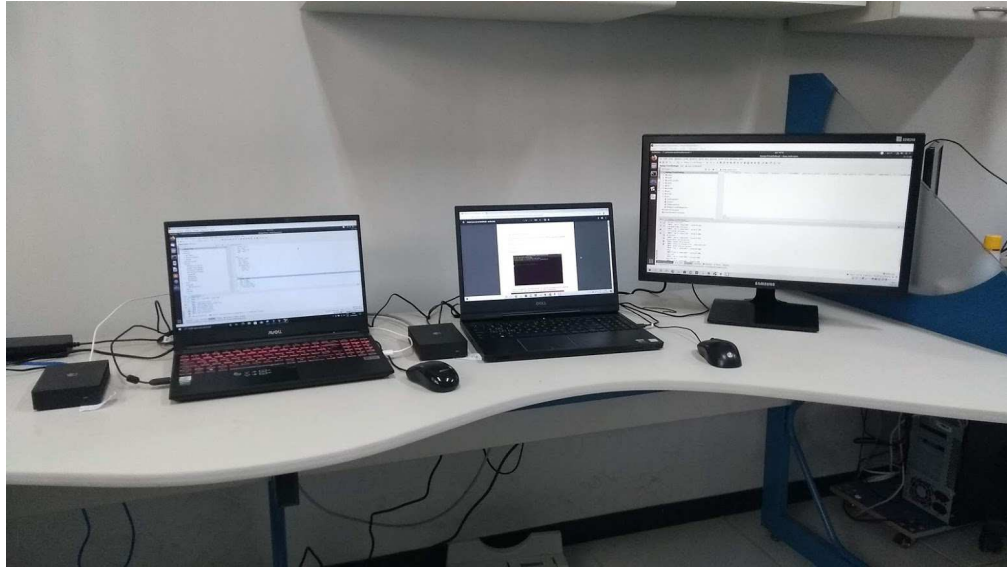


Figura 4.3: Fotografia da estrutura do Laboratório com 2 estações de trabalho utilizado no projeto piloto.

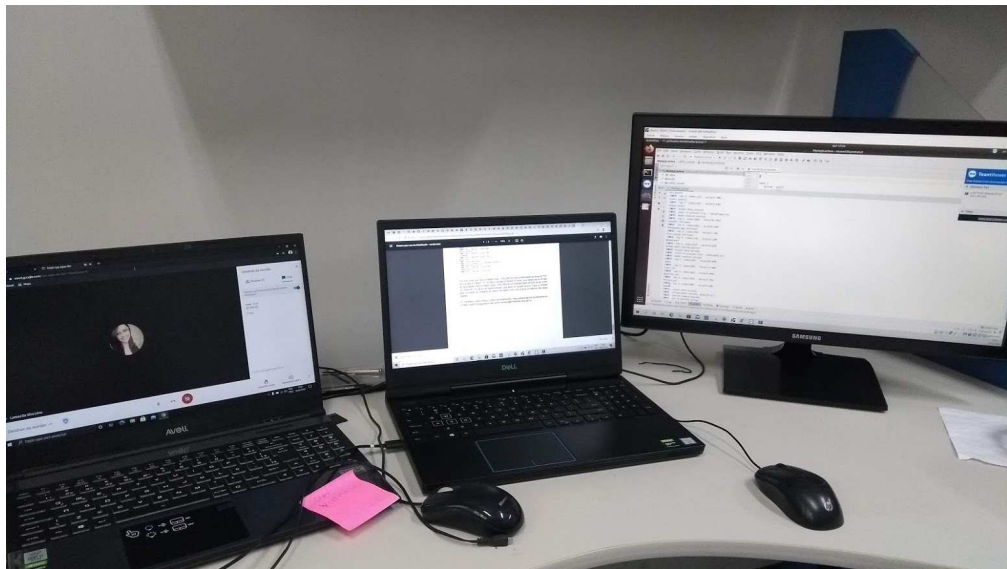


Figura 4.4: Fotografia de atividade com aluno atuando remotamente no laboratório.

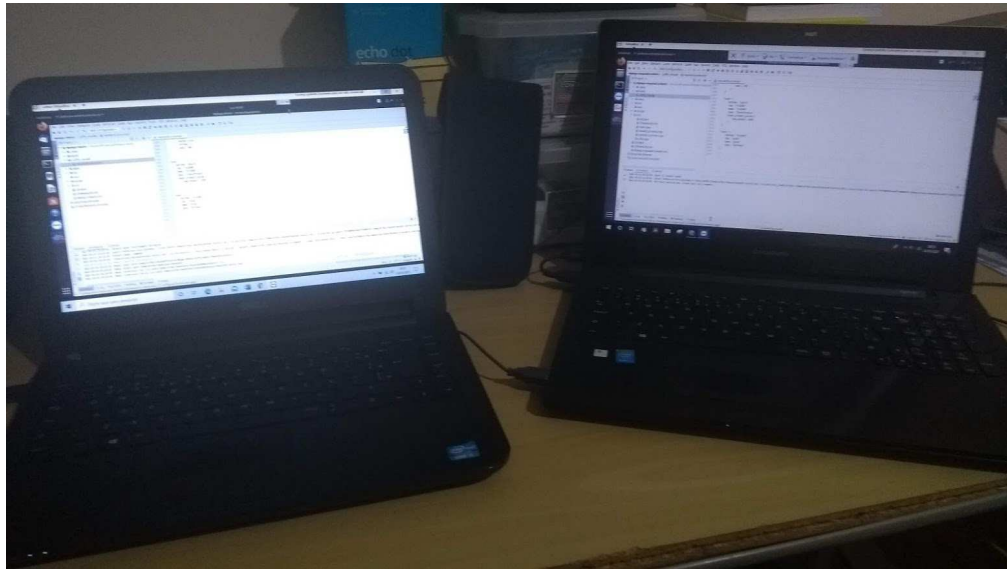


Figura 4.5: Fotografia de atividade com aluno e instrutor atuando remotamente no laboratório.

5 Conclusão

O presente trabalho descreveu as atividades do aluno Arthur de Lima Carneiro durante seu estágio no LIHM na Universidade Federal de Campina Grande. Foi apresentada uma revisão tecnológica dos equipamentos utilizados, um relatório das principais atividades realizadas: levantamento de material, estudo de funcionamento, elaboração de metodologia para laboratório remoto e aplicação de projeto piloto.

Nesse estágio diversas disciplinas cursadas durante a graduação tiveram importância. Sobre tudo as disciplinas de Programação, pois seus conceitos foram explorados em diversos momentos do projeto. Devem ser destacadas as disciplinas de Arquitetura de Sistemas Digitais e Sistemas de Aquisição de Dados e Interface que têm diversos tópicos explorados na documentação do Atlas 200 DK. Haveria um aproveitamento ainda maior se a área de Inteligência Artificial tivesse maior enfoque, haja vista o seu crescimento nos últimos anos. Entretanto, isso não apresentou como um impeditivo didático para o estagiário ou para os participantes do piloto.

Durante o período de trabalho houveram algumas dificuldades, mas a principal foi a barreira idiomática com o material encontrado. A *Huawei* é uma empresa chinesa e a maioria da documentação é redigida em mandarim. Apesar de possuir versão traduzida em inglês, há um atraso com relação as atualizações. Além disso, a comunidade de desenvolvimento é majoritariamente oriental, então as dúvidas em fóruns eram encontradas em buscadores 'não tradicionais' como *Baidu* e precisavam passar por traduções, que por vezes não eram tão satisfatórias.

A fabricante do Atlas 200 DK está investindo bastante nos produtos e isso foi um ponto positivo. Constantemente havia contato com a comunidade de desenvolvimento e o conserto de possíveis *bugs* que eram encontrados. A versão de um *toolkit* utilizado na primeira semana do estágio era a versão 20.0, ao fim do período de estágio a versão 20.3 já encontrava-se disponível. Essas constantes atualizações geravam episódios do site da documentação passar algumas horas fora do ar por estar em reforma. Por vezes, ao retornar, alguns materiais já eram considerados versões descontinuadas e sumiam, reaparecendo somente alguns dias depois.

O presente trabalho foi de bastante importância e conseguiu complementar a formação do estagiário. Ter contato com uma equipe capacitada que forneceu todo o suporte necessário foi bastante proveitoso e foi possível ter uma experiência profissional semelhante ao mercado de trabalho.

Ao término do trabalho foram alcançados todos os objetivos. Foi estudado o Atlas 200 DK, desenvolvido um material de instalação e guia de experimentos didáticos que garantem a replicabilidade do trabalho. Em virtude da pandemia da COVID-19, o modelo remoto precisou ser adotado e tornou a atividade mais versátil, pois o laboratório pode ter atividades presenciais ou remotas sem prejuízo didático.

Devido às questões de versionamento relatadas anteriormente, o estagiário montou um repositório com os arquivos utilizados na instalação e nos experimentos. Todos os materiais elaborados encontram-se nos apêndices deste trabalho.

Para trabalhos futuros sugere-se explorar as funcionalidades das câmeras integradas com o módulo Hi3559C e implantar sensores na placa de entradas e saídas de modo a gerar modelos de IA com as leituras feitas. Além disso, o Atlas 200 DK não dispõe de uma interface visual e uma aplicação que pudesse executar modelos interativamente com uma tela seria possível.

Referências

- [1] LABORATÓRIO DE INTERFACE HOMEM-MÁQUINA - DEPARTAMENTO DE ENGENHARIA ELÉTRICA. Laboratório de Interface Homem-Máquina - Departamento de Engenharia Elétrica. Disponível em: <http://www.dee.ufcg.edu.br/home/infraestrutura/laboratorios-de-pesquisa/laboratorio-de-interface-homem-maquina>. Acesso em: 17 maio. 2021.
- [2] Huawei Atlas 200 Developer Kit. [s.l.] , 2020. Disponível em: <https://support.huaweicloud.com/intl/en-us/twp-atlas200dkappc32/twp-atlas200dkappc32.pdf>.
- [3] MindStudio User Guide. [s.l.] , 2020. Disponível em: <https://support.huaweicloud.com/intl/en-us/usermanual-mindstudioc73/usermanual-mindstudioc73.pdf>.
- [4] Connecting Atlas 200 DK to Ubuntu Server_Atlas 200 DK AI developer kit (1.0.8.alpha) _Environment Deployment_Setting Up the Hardware Environment_HUAWEI CLOUD. Disponível em: https://support.huaweicloud.com/intl/en-us/dedg-A200dk_3000_c75/atlased_04_0015.html. Acesso em: 14 maio. 2021.

6 Apêndice

6.1 Instalação do Atlas 200 DK e Mindstudio

Esse material tem por objetivo auxiliar o instrutor ou aluno a realizar a instalação do ambiente de trabalho com Atlas 200 DK e Mindstudio. Todas as informações contidas foram levantadas a partir da documentação oficial do Atlas 200 DK e Mindstudio, além de tópicos de fóruns da comunidade especializada. São disponibilizados links dos portais oficiais da *Huawei*, entretanto nesse repositório no github encontram-se todos os arquivos utilizados nesse guia www.github.com/arthur-carneiro/lab-remoto-atlas200dk.

6.1.1 Material Necessário

- Atlas 200 DK (incluindo cabo USB-C e fonte de alimentação).
- Computador com sistema operacional Ubuntu 18.04 e com conexão com internet;
- Cartão micro-SD com 8 GB de armazenamento. Modelos Recomendados:
 - Samsung UHS-I U3CLASS 10 64 GB;
 - Kingston UHS-I U1 CLASS 10 64 GB;
- adaptador micro-SD para SD ou micro-SD para USB (recomendado).

6.1.2 Procedimentos

1. Instalar o Sistema Operacional Ubuntu 18.04 com idioma em inglês(US). Preferencialmente em uma máquina virtual. Siga as instruções comuns de instalação.
2. Escolha um nome de usuário e uma senha. Anotar em algum lugar, será necessário utilizar diversas vezes durante a instalação. Nesse guia de instalação o usuário será chamado de 'nome', entretanto escolha qualquer um de sua preferência.
3. Crie uma senha para o usuário root, que dará acesso a ações privilegiadas no sistema operacional. Para isso, digite o seguinte comando:

```
su root
```
4. Digite o seguinte comando para obter permissão para modificar o arquivo sudoers e em seguida acesse-o.

```
chmod u+w /etc/sudoers
vi /etc/sudoers
```
5. Ao abrir o arquivo faça a seguinte edição conforme a Figura ?? substituindo 'nome' pelo seu usuário escolhido no procedimento 2.
6. Após isso, execute o comando a seguir para desabilitar as permissões de edição do arquivo sudoers.

```
chmod u-w /etc/sudoers
```

7. Execute o comando abaixo. Será acessado o arquivo para alterar as sources do sistema operacional para que sejam acessados os servidores da China.

```
vi /etc/apt/sources.list
```

Apague o texto do arquivo e substitua pelo seguinte:

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe
multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe
multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe
multiverse
```

Após isso, volte ao terminal e execute o comando a seguir:

```
apt-get update
```

E, em seguida, retorne para o usuário nome com o seguinte comando:

```
exit
```

8. Já com o usuário nome, execute o comando a seguir para criar a pasta .pip e criar um arquivo de texto:

```
mkdir $HOME/.pip/
vi $HOME/.pip/pip.conf
```

Com o arquivo criado, cole o seguinte texto:

```
[global]
timeout = 6000
index-url = http://mirrors.aliyun.com/pypi/simple/
trusted-host = mirrors.aliyun.com
```

Salve o arquivo e retorne ao terminal.

9. Execute o seguinte comando para instalar diversos pacotes para o ambiente de desenvolvimento.

```
sudo apt-get install -y gcc g++ make cmake unzip zlib1g
sudo zlib1g-dev libsqlite3-dev openssl libssl-dev
sudo libffi-dev pciutils net-tools
sudo g++-5-aarch64-linux-gnu libblas-dev
sudo gfortran libblas3 libopenblas-dev
```

10. Agora será instalado o ambiente de desenvolvimento para a linguagem Python. Primeiramente retorne para a pasta Home:

```
cd $HOME
```

Em seguida execute os comandos a seguir para baixar e extrair a pasta:

```
wget https://www.python.org/ftp/python/3.7.5/Python-3.7.5.tgz
tar -zxvf Python-3.7.5.tgz
```

Entre na pasta gerada e, em seguida, execute os comandos de instalação e configuração:

```
cd Python-3.7.5
./configure --prefix=/usr/local/python3.7.5 --enable-shared
make -j8
sudo make install
```

Execute os comandos a seguir para gerar o link entr as variáveis de ambiente e as bibliotecas instaladas:

```
sudo cp /usr/local/python3.7.5/lib/libpython3.7m.so.1.0 /usr/lib
sudo ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7
sudo ln -s /usr/local/python3.7.5/bin/pip3 /usr/bin/pip3.7
sudo ln -s /usr/local/python3.7.5/bin/python3 /usr/bin/python3.7.5
sudo ln -s /usr/local/python3.7.5/bin/pip3 /usr/bin/pip3.7.5
```

11. Execute o comando a seguir:

```
vi ~/.bashrc
```

O .bashrc irá abrir em um arquivo de texto. Copie o trecho a seguir e cole ao fim do texto:

```
export LD_LIBRARY_PATH=/usr/local/python3.7.5/lib:$LD_LIBRARY_PATH
export PATH=/usr/local/python3.7.5/bin:$PATH
```

Após isso, salve o arquivo e retorne ao terminal. Em seguida, execute as atualizações com o comando a seguir:

```
source ~/.bashrc
```

12. Execute o comando a seguir para instalar bibliotecas para o funcionamento do python:

```
pip3.7.5 install attr psutil decorator numpy protobuf==3.11.3 scipy sympy cffi --user
pip3.7.5 grpcio grpcio-tools requests --user
```

13. Acesse o endereço a seguir <https://www.hiascend.com/en/software/cann> e faça download dos seguintes pacotes:

```
Ascend-cann-toolkit_20,1rc1.alpha001_linux-x86_64.run
Ascend-cann-toolkit_20,1rc1.alpha001_linux-aarch64.run
```

14. Execute o comando abaixo:

```
chmod +x *.run
```

E em seguida instale os pacotes que foram baixados com os comandos a seguir:

```
Ascend-cann-toolkit_20,1rc1.alpha001_linux-x86_64.run --install
Ascend-cann-toolkit_20,1rc1.alpha001_linux-aarch64.run --install
```

E instale os pacotes abaixo:

```
sudo apt-get install -y g++-aarch64-linux-gnu g++-5-aarch64-linux-gnu
```

15. Acesse o endereço a seguir, procure e faça download do Mindstudio 2.0.0 beta2 na parte "version history". O versionamento é bastante importante, pois há um "casamento" das versões do Mindstudio com a versão 20.1rc1.

<https://www.hiascend.com/en/software/mindstudio/download>

16. O MindStudio será baixado em modo compactado, extraia os arquivos para que possa ser feita a instalação do software. Execute o comando a seguir:

```
tar -zxvf MindStudio_2.0.0-beta2_ubuntu18.04-x86_64.tar.gz
```

17. A pasta MindStudio-ubuntu será criada, entre na mesma e execute os comandos a seguir:

```
cd MindStudio-ubuntu/bin  
./Mindstudio.sh
```

Dessa forma será aberto o menu de instalação do software. Pode ser que sejam pedidos alguns pacotes complementares, baixe-os com o seguinte comando:

```
sudo apt-get -y install xterm openjdk-8-jdk fonts-wqy-zenhei fonts-wqy-microhei  
sudo apt-get -y install fonts-arphic-ukai fonts-arphic-uming  
/usr/local/python3.7.5/bin/ pip3 install --user coverage gnureadline pylint  
/usr/local/python3.7.5/bin/ pip3 install --user matplotlib xlrd PyQt5==5.14.0
```

18. Repita o passo 11 para o seguinte endereço:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64  
export PATH=$JAVA_HOME/bin:$PATH
```

19. Após isso, repita o procedimento de abertura do MindStudio e escolha "/home/nome/Ascend/ascend-toolkit/20.1rc1" como toolkit location.

O Mindstudio terá sido instalado com sucesso caso a janela seja semelhante a Figura 6.3.

20. A partir desse procedimento será feita a instalação do Ubuntu do Atlas 200 DK. São necessários os downloads de 4 arquivos, sendo 2 scripts e 2 toolkits. Além da imagem .iso do Ubuntu 18.04 server ARM.

No site oficial do Ubuntu, <https://ubuntu.com/>, faça download de qualquer uma das seguintes versões:

- ubuntu-18.04.4-server-arm64.iso
- ubuntu-18.04.5-server-arm64.iso

No endereço a seguir: <https://www.hiascend.com/zh/software/cann>, faça o download do arquivo:

- Ascend-cann-minirc_20.1.rc1_ubuntu18.04-aarch64.zip

E no seguinte: <https://www.hiascend.com/hardware/firmware-drivers>, faça o download do arquivo:

- A200dk-npu-driver-20.1.0-ubuntu18.04-aarch64-minirc.tar.gz

No repositório citado no início desse material encontram-se os seguintes arquivos:

- make_sd_card.py
- make_ubuntu_sd.sh.

Copie-os e entre na pasta make_sd cole nessa pasta todos os arquivos citados nesse procedimento. Devem estar conforme a Figura 6.1

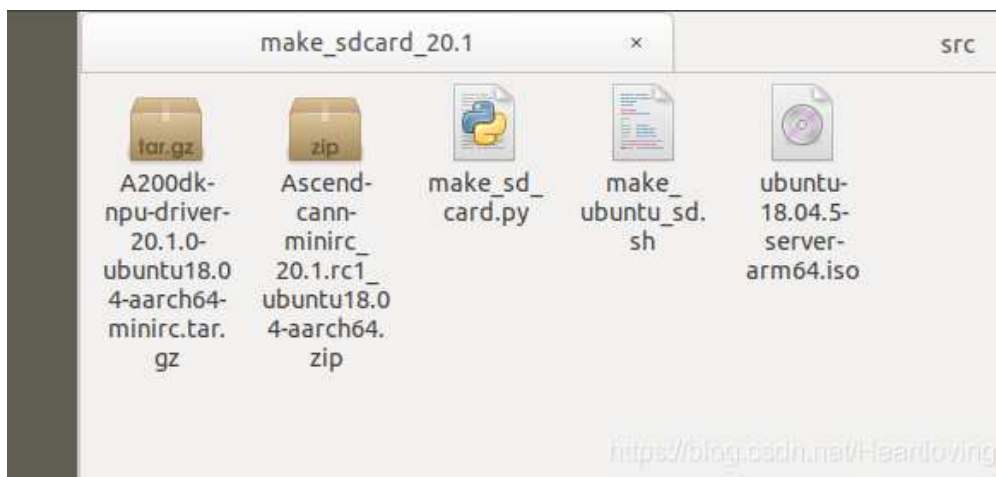


Figura 6.1: Arquivos necessários para instalação do Atlas 200 DK.

21. Com todos os arquivos prontos, entre em modo root e, em seguida, instale os pacotes a seguir:

```
su root
pip3.7.5 install pyyaml
apt-get install qemu-user-static binfmt-support python3-yaml gcc-aarch64-linux-gnu
apt-get install g++-aarch64-linux-gnu
```

22. Navegue pelo terminal e insira o cartão de memória no computador. Espere ser reconhecido pela máquina e em seguida digite o seguinte comando:

```
fdisk -l
```

O resultado será a listagem de todos os dispositivos de memória conectados ao computador. Identifique o endereço do cartão de memória (normalmente é /dev/sdb ou /dev/sdc). Após identificar execute o comando abaixo para iniciar a instalação.

```
python3.7.5 make_sd_card.py local /dev/sdb
```


Caso o endereço seja distinto, substitua 'sdb' pelo encontrado em seu computador. Espere de 5 a 7 minutos para a resposta de que a instalação teve êxito. Caso haja algum erro, sugiro recortar todo o texto de "make_sd_card.py", salvar o arquivo, depois colar e salvar novamente. Caso esteja tudo certo, possivelmente após fazer isso irá funcionar nos conformes.

23. Ao fim da instalação, retire o cartão de memória do computador e insira no Atlas 200 DK. Aguarde a instalação interna do dispositivo. Após o acendimento dos 4 LEDs, conecte-o via USB ou *Ethernet* no computador. Em seguida busque o seguinte comando:

```
ifconfig
```

Deverá ser encontrado um endereço semelhante a ens160u2. Em seguida, digite o comando a seguir:

```
vim /etc/netplan/01-network-manager-all.yaml
```

E substitua os valores encontrados aos seguintes:

```
network:
version: 2
renderer: NetworkManager
ethernets:
  ens160u2:
    dhcp4: no
    addresses: [192.168.1.223/24]
    gateway4: 192.168.0.1
    nameservers:
      addresses: [255.255.0.0]
```

Em seguida digite netplan apply. Confira em ifconfig se ens160u2 agora dispõe de um endereço de IP. Isso será estritamente necessário para conseguir conectar com o Atlas 200 DK via SSH.

24. Realize o SSH no Atlas, a senha default é Mind@123:

```
ssh HwHiAiUser@192.168.1.2
```

Encontre o arquivo libascend.so com os comandos a seguir:

```
su root
find / -name libascendcl.so
```

Identifique o arquivo /home/HwHiAiUser/Ascend/acclib/lib64/libascendcl.so. Copie o caminho até o arquivo. Execute os comandos abaixo:

```
cd /etc/ld.so.conf.d
vim mind_so.conf
```

No arquivo que abrir, cole o caminho do arquivo libascendcl.so, salve e retorne ao terminal. Em seguida, digite ldconfig para salvar as alterações.

25. Nesse momento o Atlas 200 DK e o ambiente do MindStudio estão configurados. Bons estudos!

6.2 Experimento 1 - Conversão de modelos Caffe com Mindstudio

6.2.1 Apresentação da Atividade

O Atlas 200 DK é um equipamento de IA voltado para desenvolvimento de aplicações de alta performance utilizando o chip Ascend 310. Esse processador é utilizado exclusivamente para inferência de modelos pré-treinados, aplicações que exijam treinamento utilizam-se do chip Ascend 910. O Atlas 200 DK pode ser utilizado para pesquisa científica, soluções industriais ou como ferramenta didática para fins educacionais.

O Mindstudio é a IDE utilizada pelos equipamentos produzidos pela Huawei para as aplicações em IA. Neste ambiente é possível desenvolver, revisar, converter modelos em diversos cenários.

A partir disso, uma das aplicações do MindStudio é a possibilidade de converter modelos desenvolvidos em outros frameworks Caffe ou Tensorflow. Esses modelos são convertidos para o Mindspore, que é o framework desenvolvido para as soluções em IA da Huawei,

Nesse experimento o aluno irá conhecer o ambiente do Mindstudio, abrir um projeto da rede resnet-50, converter um modelo de Caffe para Mindspore, ajustar dados de entrada para o modelo, compilar a aplicação e realizar o deploy da mesma no Atlas 200 DK.

Após realizar todos os procedimentos, no tópico de discussões o aluno será convidado a realizar uma etapa do experimento que permita uma análise mais crítica do experimento.

6.2.2 Materiais

Para realizar o presente experimento os seguintes itens são necessários:

- Atlas 200 DK com Cartão SD configurado;
- Computador com sistema operacional Ubuntu 18.04, podendo ser nativo ou Máquina Virtual, com 4 GB de memória RAM e configurado com os toolkits CANN Ascend (Ver procedimento 14 na seção 6.1)
- Mindstudio 2.0.0 (beta2) instalado.
- Teamviewer instalado.

Caso o experimento seja realizado pelo aluno em formato remoto, o mesmo deve dispor somente do Teamviewer instalado no computador para estabelecer conexão com o laboratório. Todos os equipamentos acima listados devem estar instalados no LIHM pelo instrutor. No repositório www.github.com/arthur-carneiro/lab-remoto-atlas200dk estão todos os arquivos utilizados nesse experimento.

6.2.3 Procedimentos

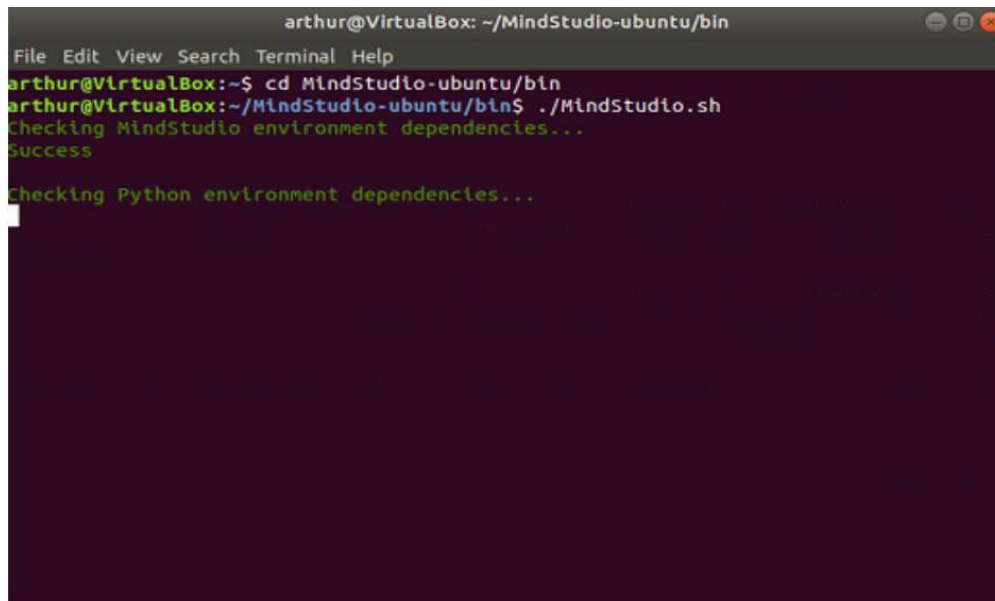
1. Primeiramente iremos abrir o MindStudio. Para isso, abra um terminal (ctrl+shift+T) e execute os seguintes comandos:

```
cd MindStudio-ubuntu/bin
```

Em seguida:

```
./Mindstudio.sh
```

As saídas devem estar de acordo com a Figura 6.2 a seguir.



```
arthur@VirtualBox: ~/MindStudio-ubuntu/bin
File Edit View Search Terminal Help
arthur@VirtualBox:~$ cd MindStudio-ubuntu/bin
arthur@VirtualBox:~/MindStudio-ubuntu/bin$ ./MindStudio.sh
Checking MindStudio environment dependencies...
Success
Checking Python environment dependencies...
```

Figura 6.2: Saídas para os comandos do terminal

2. Haverá o carregamento das dependências e em seguida uma janela se abrirá. Clique em "Create New Project" caso seja a primeira vez que utiliza no MindStudio. O botão está destacado em amarelo na Figura 6.3.

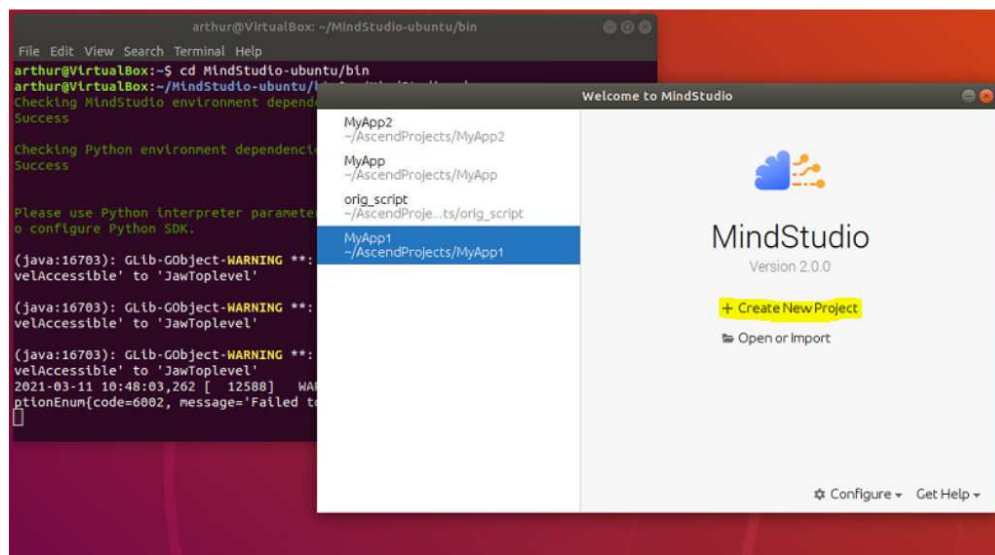


Figura 6.3: Janela de inicial do MindStudio.

Caso contrário vá em "File" → "New" → "Project..." no menu que fica no topo da tela, como mostra a Figura 6.4.

3. A tela de abertura do projeto será criada e você deve nomeá-lo, como mostrado na Figura

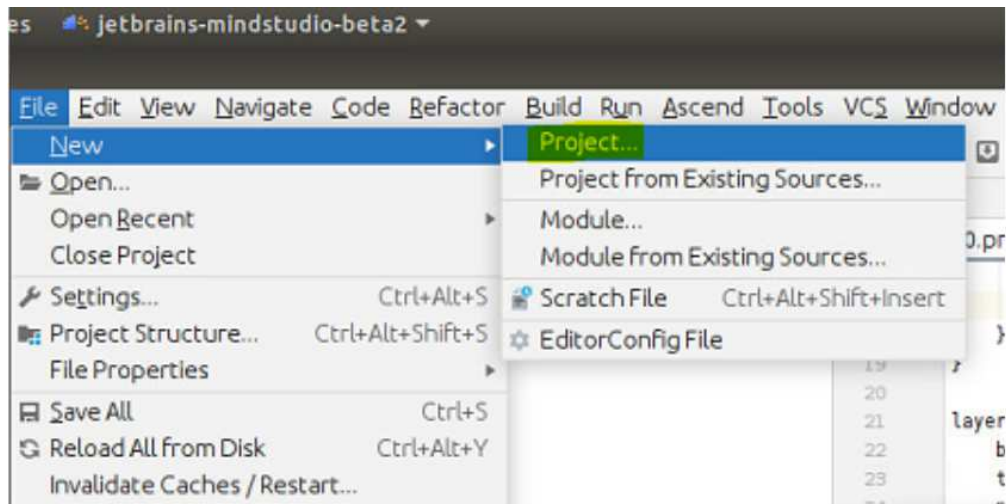


Figura 6.4: Modo alternativo para abertura de novo projeto no MindStudio.

6.5. Para fins de diferenciação vamos chamar de "MyApp-nome". Onde 'nome' escreva seu primeiro nome ou sobrenome. Selecione "Ascend App" (à esquerda) e clique em *Next*.

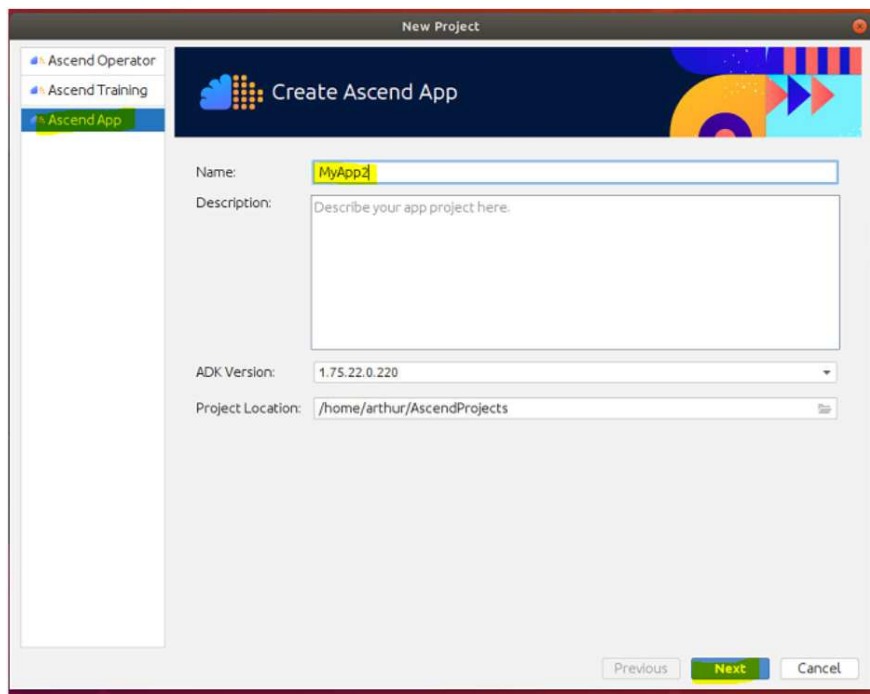


Figura 6.5: Tela de abertura de novo projeto no Mindstudio.

4. Selecione "Sample Resnet-50" e clique em "*Finish*" como indicado na Figura 6.6. Será criada uma pasta com uma estrutura com diversas pastas encadeadas.
5. Clique em "MyApp-nome" com o botão direito e aponte para New → Directory, como indicado na Figura 6.7 e clique com o botão esquerdo. Nomeie esse diretório como "*caffe_model*".

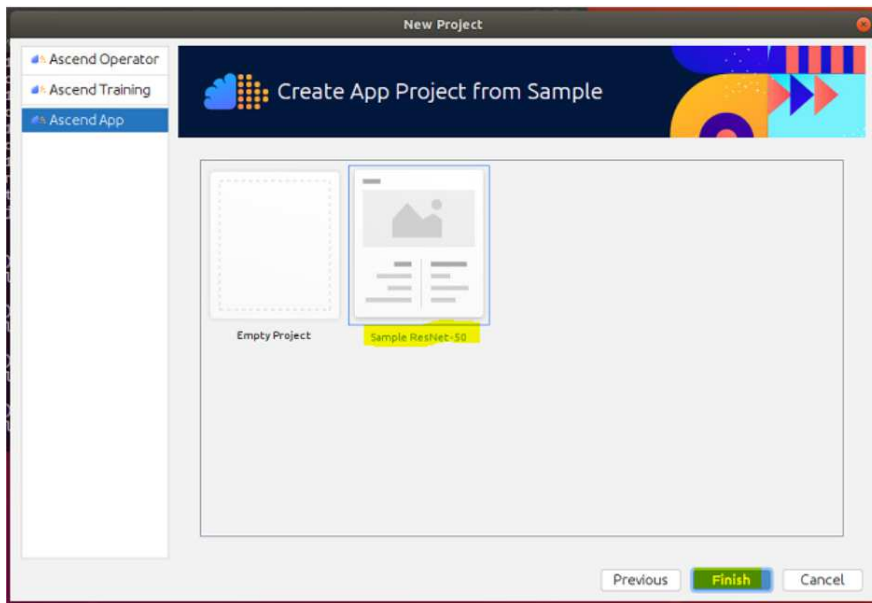


Figura 6.6: Tela de para selecionar banco de dados pré-existente.

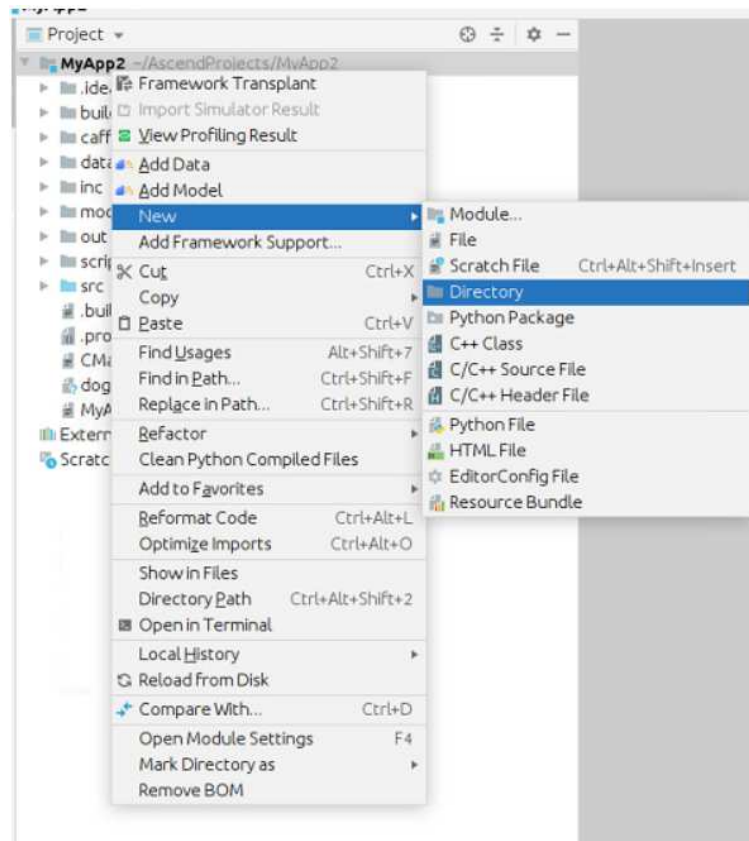


Figura 6.7: Criação de nova pasta no projeto.

6. Minimize o MindStudio e vá para a área de trabalho. Depois clique com o botão direito em *“resnet50.prototxt”* e, em seguida, clique em *“copy”*.

7. Retorne ao MindStudio e clique com o botão direito na pasta `caffe_model`, como indicado na Figura 6.8, em seguida clique em “paste”. Clique em “refactor” para confirmar. Esse arquivo possui as informações acerca da rede neural que será usada no modelo.

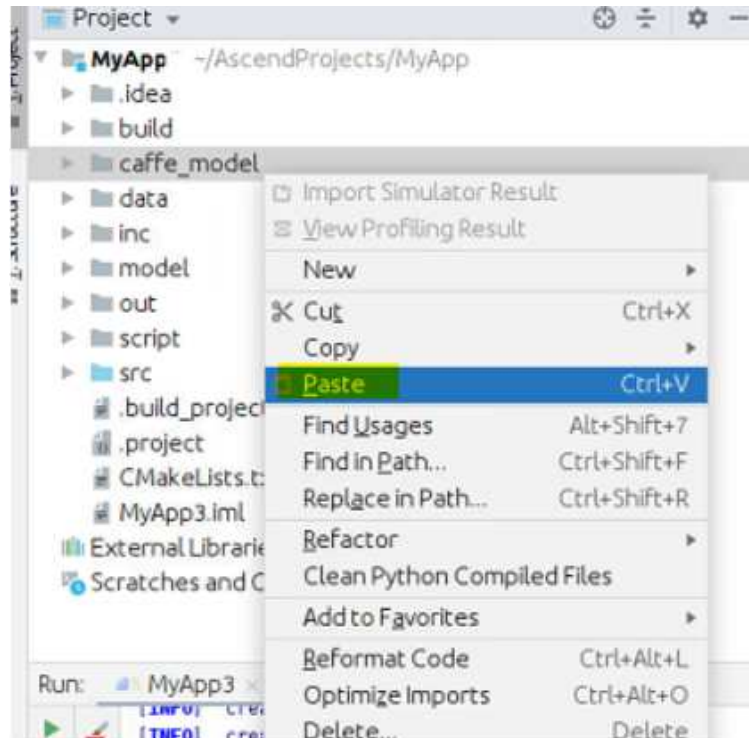


Figura 6.8: Transferência do Arquivo `resnet50` para a pasta `caffe_model`.

8. Em seguida, no menu superior clique em `Ascend` → `Model Converter`. Abrirá uma janela semelhante a da Figura 6.9 Em `Model File` selecione `'resnet50.prototxt'` na pasta `caffe_model`. Em `Weight File` selecione o arquivo `"resnet50.caffemodel"` que está na pasta `"Desktop"`.
Em `Model Name`, `Target SoC Version` e `Input Format` não faça alterações. Em `Input Nodes - Type` selecione `FP16`. Clique em `"next"`.
9. Caso surja uma janela alegando que o modelo já existe, clique em `replace`.
10. Retire a seleção de `Data Preprocessing` (a opção deve ficar cinza, assim como `Input Node`). Clique em `next` e em seguida em `Finish`.
11. O modelo será convertido e deve aparecer no canto inferior da tela em verde a seguinte mensagem: `"Model converted successfully"`. Obs: A mensagem pode demorar alguns instantes a aparecer. Aguarde um pouco.
12. Clique com o botão direito em `MyApp-nome` e em seguida `"Add Model"` como indicado na Figura 6.10. Na pasta `modelzoo/resnet50/device/` selecione `"resnet50.om"`, como indicado na Figura 6.11. Será criada uma pasta `"model"` em que esse arquivo estará armazenado no projeto.

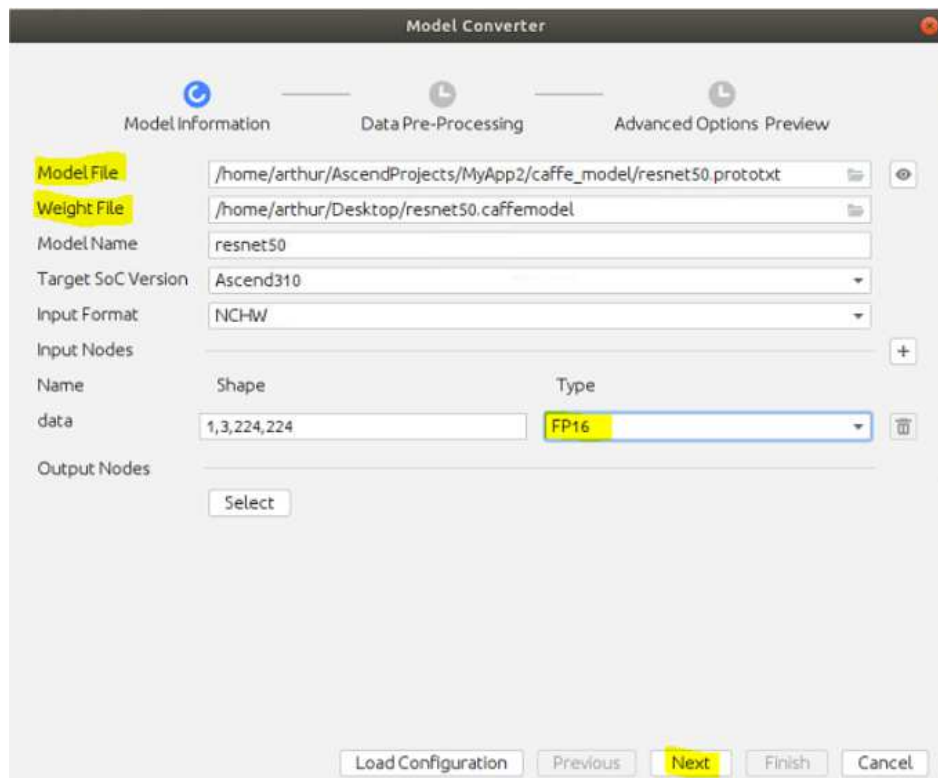


Figura 6.9: Opções de conversão de modelo de Caffé para Mindspore.

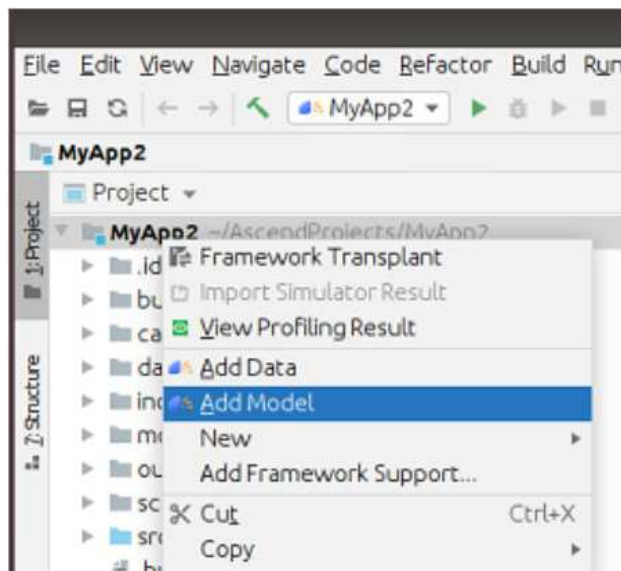


Figura 6.10: Opção para adicionar modelo Mindspore ao projeto.

13. Selecione as fotos que estão na área de trabalho, clique no botão direito e selecione “copy”. No canto esquerdo da tela clique em “Files”, navegue pra AscendProjects/MyApp-nome/data. Clique com o botão direito e selecione “paste”.
14. Abra uma janela do terminal e digite o seguinte comando:

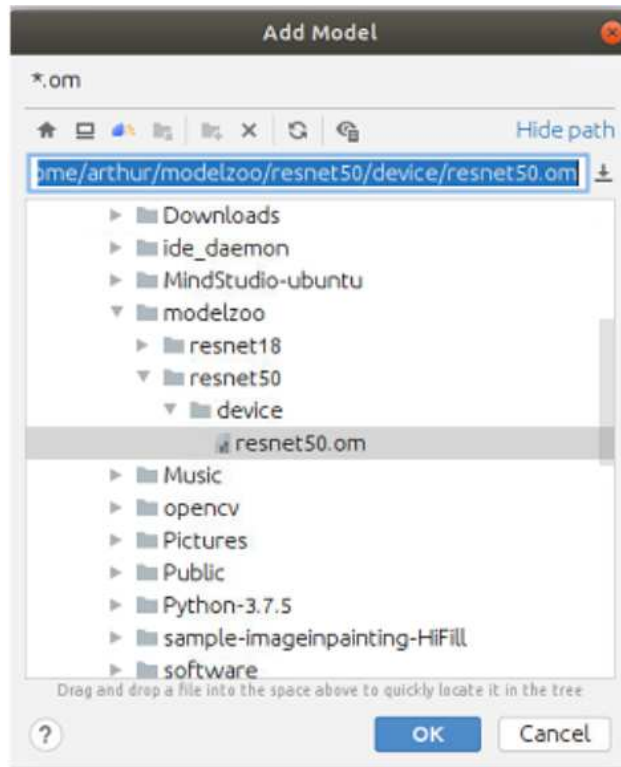


Figura 6.11: Localização do modelo convertido para adicionar ao projeto.

```
cd ~/AscendProjects/MyApp-nome/data
```

Em seguida, digite:

```
python3.7.5 ../script/transferPic.py
```

Os comandos irão converter as imagens em /data do formato *.jpeg para *.bin. Para ver se o script foi executado corretamente a saída deve ser semelhante à observada na Figura 6.12.

15. Copie os arquivos “json.hpp”, “utils.cpp”, “model_process.cpp” e “sample_process.cpp” que estão na área de trabalho. Navegue a partir de “Files”, vá para a pasta AscendProjects/MyApp-nome/src, clique com o botão direito e “paste”. Clique em “Apply this action to all files and folders” e “Replace”.
16. Copie o arquivo “class_index.json” na área de trabalho. Navegue a partir de “Files”, vá para a pasta AscendProjects/MyApp-nome/out, clique com o botão direito e selecione “paste”.
17. Expandindo a pasta data, que está no endereço AscendProjects/MyApp-nome/data/, você visualizará agora 8 arquivos, sendo 4 imagens jpg.
18. No menu superior do MindStudio clique em Build → Edit Build Configuration. As opções devem ser as seguintes:

- Configuration Name: Build-Configuration


```
arthur@VirtualBox: ~/AscendProjects/MyApp2/data
File Edit View Search Terminal Help
arthur@VirtualBox:~$ clear
arthur@VirtualBox:~$ cd ~/AscendProjects/MyApp2/data
arthur@VirtualBox:~/AscendProjects/MyApp2/data$ python3.7.5 ../script/transferPic.py
start to process image dog2_1024_683.jpg...
process image dog2_1024_683.jpg successfully
start to process image dog1_1024_683.jpg...
process image dog1_1024_683.jpg successfully
start to process image gato.jpg...
process image gato.jpg successfully
start to process image Puppy.jpg...
process image Puppy.jpg successfully
4 images in total, 4 images process successfully, 0 images process failed
arthur@VirtualBox:~/AscendProjects/MyApp2/data$
```

Figura 6.12: Localização do modelo convertido para adicionar ao projeto.

- Target: SOC
- Target OS: Linux
- Target Architecture: aarch64

Em seguida clique em Build. Se o modelo foi construído corretamente deverá aparecer as mensagens a seguir na janela de log do MindStudio:

Information:[100%] Built target main.

Information: MyApp–Nome ou Sobrenome build successfully.

19. No menu superior clique em Run. Nessa opção o modelo convertido será enviado ao Atlas 200DK e serão realizadas as inferências sobre os modelos criados.
20. O modelo foi carregado e referenciado corretamente caso a saída seja semelhante a observada na Figura 6.13

```

2021-03-11 16:33:12 - [INFO] Assigning execute permission to run.sh on the remote
/home/HwHiAiUser/HIAI_PROJECTS/workspace_mind_studio/MyApp3_9f210d93/out/run.sh: 1
sudo: no tty present and no askpass program specified
[INFO] acl init success
[INFO] open device 0 success
[INFO] create context success
[INFO] create stream success
[INFO] get run mode success
[INFO] load model ../model/resnet50.om success
[INFO] create model description success
[INFO] create model output success
[INFO] start to process file:../data/dog1_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[161] - value[76.76%]
basset
[INFO] top 2: index[162] - value[15.48%]
beagle
[INFO] top 3: index[167] - value[3.85%]
English_foxhound
[INFO] top 4: index[163] - value[2.16%]
bloodhound
[INFO] top 5: index[166] - value[1.17%]
Walker_hound
[INFO] output data success
[INFO] start to process file:../data/dog2_1024_683.bin
[INFO] model execute success
[INFO] top 1: index[267] - value[93.60%]
standard_poodle
[INFO] top 2: index[266] - value[4.11%]

```

Figura 6.13: Log de execução do projeto com identificação das imagens.

Para a imagem dog1_1024_683 há uma probabilidade de cerca de 76% de que seja a imagem 161 do banco de dados utilizado no treino, que nesse caso é um cão da raça basset. Para a imagem dog2_1024_683 há uma probabilidade de 93% de ser a foto de índice 267 do banco de dados treinado, que seria um poodle comum. Fique a vontade para comparar as imagens do banco de dados com uma busca na internet das raças citadas.

21. Parabéns, você cumpriu o roteiro do experimento. Caso possua alguma dúvida entre em contato com seu instrutor.

6.2.4 Discussões

O roteiro desse experimento leva o aluno a interagir com o ambiente do Mindstudio e a interação com o Atlas 200 DK. Foram feitas mudanças estruturais no projeto, adição de modelos externos. O mindstudio permite avaliação de arquitetura de redes neurais, visualização de camadas escondidas entre muitas outras funcionalidades.

O experimento torna-se mecânico e pouco crítico se não houver uma reflexão do que foi desenvolvido. Por isso, nesse momento você pode avaliar a rede neural com uma imagem estranha ao modelo. Siga os passos a seguir:

1. Abra o arquivo class_index e identifique entre todas as classes alguma do seu interesse.

2. Pesquise no google imagens uma foto da classe escolhida. Faça download da imagem e cole na pasta "data" do modelo. Semelhante ao procedimento 13.
3. Abra o arquivo "sample_process.cpp" do seu projeto e procure no código "string TextFile" e digite o nome do seu arquivo semelhante aos outros que estão no trecho do código. Ou seja "../data/(nome-da-foto).bin"
4. Repita o procedimento 14 do projeto.
5. Execute o projeto e veja a avaliação do modelo para aquela imagem. Os resultados foram bons? Consulte seu instrutor para refletir a respeito se houve uma boa ou má avaliação.

6.3 Experimento 2 - Acesso via SSH e Execução por linha de comando

6.3.1 Apresentação da Atividade

Nessa atividade será utilizado o mesmo modelo do experimento anterior. Entretanto não será dessa vez utilizado o MindStudio. É necessário que o Experimento 1 tenha sido realizado com sucesso para poder iniciar essa atividade.

O objetivo aqui é demonstrar que o Atlas 200 DK é um dispositivo que pode ser utilizado além de uma bancada, pode ser aplicado em dispositivos com mobilidade. Como pode ser conectado à rede e acessado via terminal é possível executar modelos remotamente, obter respostas e tomar decisões a partir dos resultados.

6.3.2 Materiais

- Atlas 200 DK com Cartão SD configurado;
- Computador com conexão com a mesma rede do Atlas 200 DK.

6.3.3 Procedimentos

1. Confira sua conexão com a internet e o IP utilizado por seu Atlas 200 DK, nesse experimento consideraremos 192.168.1.2, que é um dos endereços utilizados na configuração inicial.
2. Realize o SSH no Atlas. A senha inicial é Mind@123. Execute os comandos a seguir:
`ssh HwHiAiUser@192.168.1.2`
3. Navegue pelo terminal para a seguinte pasta:
`cd Ascend-Projects/mindstudio`
4. Localize na pasta com o comando `ls` os entre projetos existentes no Atlas 200 DK o que possua MyApp-nome, o mesmo que utilizou no Experimento 1. Entre na pasta utilizando o comando `cd`
`cd MyApp-nome`
5. Executando o comando de listar os arquivos serão encontradas as pastas `data`, `model`, `out` e `src`. Navegue para a pasta `out`.
`cd out`
Em seguida liste os arquivos existentes.
6. Execute o arquivo `run.sh`, com o comando abaixo.
`./run.sh`
7. A saída do comando será exatamente igual a do procedimento 20 no experimento anterior que é observada na Figura 6.13. Na prática, foi executado o mesmo modelo do mesmo experimento, entretanto tudo agora foi feito utilizando somente a estrutura do Atlas, sem qualquer intervenção do MindStudio.

6.3.4 Discussões

Apesar do experimento 2 possuir bem menos procedimentos quando comparado ao anterior, ele possui um valor didático interessante para o estudo do Atlas 200 DK. A partir do que foi proposto é possível afirmar que basta que tenhamos acesso ao equipamento via rede que é possível executar ações a partir do mesmo.

Sem utilizar-se do MindStudio é possível desenvolver uma aplicação no MindSpore e descarregar no Atlas 200 DK via rede e ainda executá-la. Dessa forma é possível verificar a versatilidade do Atlas 200 DK para os mais diversos cenários em que um dispositivo de inteligência artificial pode ser aplicado.