

AI の基盤としての半導体の 理解に向けての デザインハッカソン

2025年1月17日

後藤 隆男 新田見瑞穂 天野 英晴 御手洗勇輝 于 嘉偉
潘 宇軒 小菅 敦丈 池田 誠

東京大学大学院附属システムデザイン研究センター

半導体分野の人材不足

- 巨大Fabへの投資が行われているが、依然として人材不足
 - 半導体に興味を持つ学生が少ない
 - AIは大人気
- AI技術は半導体に支えられているのに、そこまで目が行かない
AI技術を入口にして、半導体が支えていることに気づいてもらう



東大半導体デザインハッカソン

Agile-Xプロジェクトで開発

Agile PROJECT

Agile-X プロジェクト

東京大学『Agile-X～革新的半導体技術の民主化拠点』プロジェクト

文部科学省は、「次世代X-nics半導体創生拠点形成事業」を開始しました。

カーボンニュートラル2050やデジタル社会の実現、経済安全保障の確保に向けて重要な役割を果たす革新的半導体集積回路の創生を目的とし、我が国の強みを活かした研究開発及び人材育成の中核的なアカデミア拠点形成を行う事業です。

その1つの拠点として、東京大学大学院工学系研究科附属システムデザイン研究センター (d.lab)の提案「Agile-X～革新的半導体技術の民主化拠点（以下、本拠点）」が採択されました。

本拠点は、2022年度から31年度までの足かけ10年間で、世界に大規模集積回路(LSI)の民主化をもたらすことを目標とします。

多様なアプリケーションに最適な半導体チップを迅速に活用しイノベーションを興すことが、新しく強い産業の創成と発展につながります。

そのためには、半導体集積回路が「特別な」ものから「誰にでも容易に実現できる」ようにすることが必要です。これを半導体技術の民主化と呼びます。

民主化を実現する基幹技術（Agile技術）を整備し、プラットフォームとして研究に展開し、研究成果・産業への波及効果・高度人材の育成を狙います。

関連研究① AIエッジコンテスト

https://www.meti.go.jp/policy/mono_info_service/joho/aiedge/index.html

- 経済産業省、NEDOによる
- 革新的なAIエッジコンピューティングの実現に向けて、優れた技術・人材・アイデアを発掘し、新たな人材の当該分野への参画を促すためのコンテスト
- 東大半導体デザインハッカソンはこの2回目参考になっている
- 第6回(2022)の課題
 - (開発) 車両前方の画像および全方位の点群データから、**3D**物体検出を行うアルゴリズムを作成
 - (実装) 開発したアルゴリズムを、**RISC-V**を搭載したプラットフォームに実装
 - 投稿41件、参加271人
 - 多くのスポンサーによる手厚い賞金 (1位は100万円 + 各種クーポン)

関連研究② 相磯杯FPGA Design Competition

<https://wp.rs.cs.okayama-u.ac.jp/design-contest-aiso11/>

- Reconf研、岡山大他関連大学協賛
- ICFPT、HEARTなど国際学会と連携、最近はICCEと連携
- FPGAを搭載した自動走行車がカメラからの画像のみで自動走行し、信号、歩行者、障害物回避などのタスクを実行する
- 10チーム前後



関連研究③ Intel InnovateFPGA design contest

<https://www.intel.co.jp/content/www/jp/ja/products/docs/programmable/innovate-fpga-design-contest.html>

- FPGA をベースにしたサステナビリティをテーマとするプロジェクトの開発
- サンゴ礁の回復、ソーラーコンバータのクラウドベースの管理などかなり大きなプロジェクト
- FPGA CloudConnectivity Kit を使ったクラウドとエッジの連携が鍵
- FPGA自体はCycloneでさほど強力ではないが、クラウドと連携して大きなプログラムを動かす
- 2022年に開催、260人が参加、それ以降は行われていない

関連研究のまとめとモチベーション

- 三つ共、特色を持った興味深いコンテストで、半導体設計分野の育成にも効果がある
- どれもハードウェアの設計経験のある程度必要とし、設計時間もかなり長い
- AIに興味を持つ学生を半導体分野に呼び込む目的には使えない



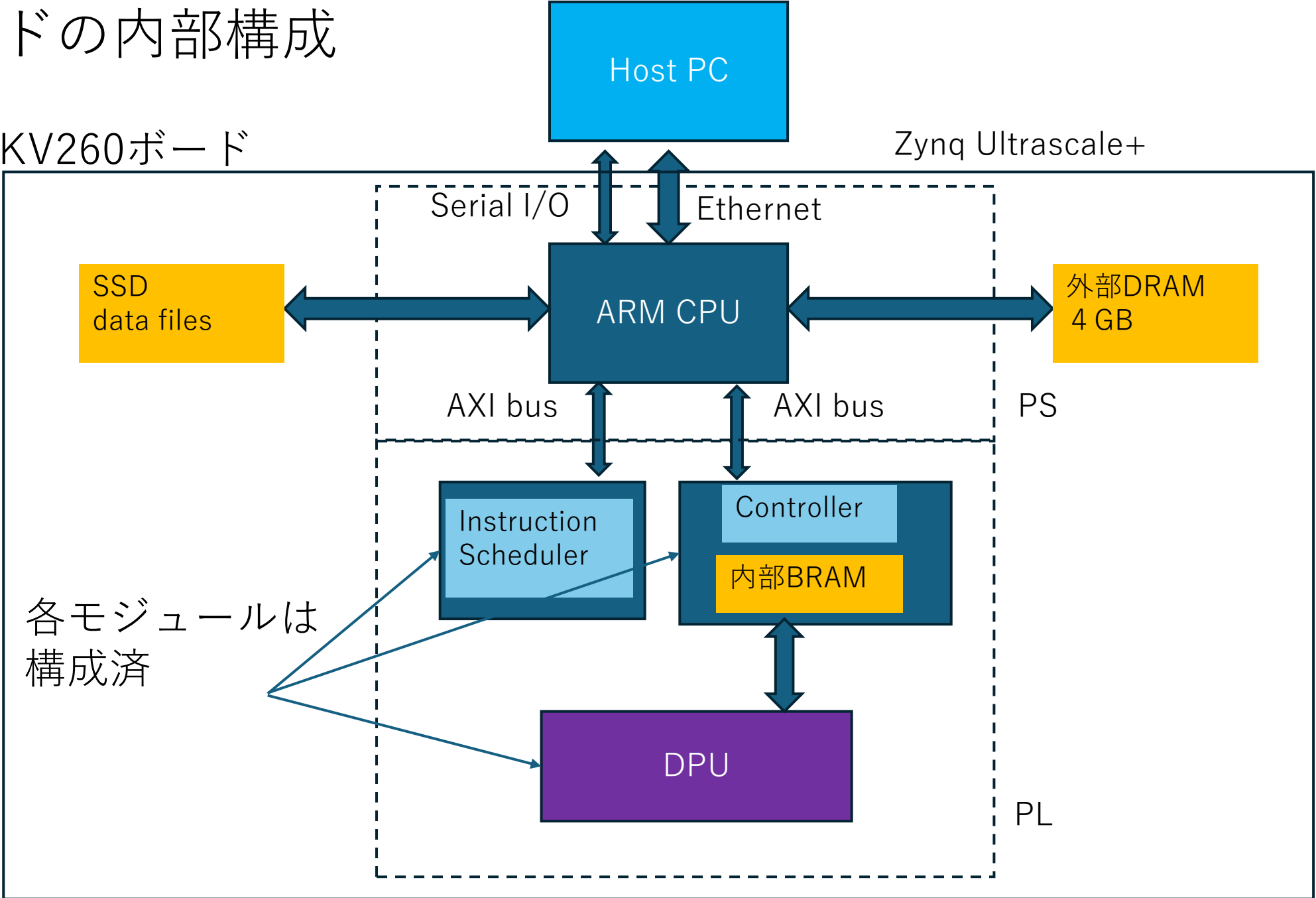
- 今回のデザインハッカソンのポイント
 - ハードウェア記述言語、Vivadoによる配置配線などを使わない
 - C++とVitis-AI環境のみでYolo v3を動作させる
 - ハードウェアの詳細を知らない状態で最適化
 - もちろんHLSをじかに使ったりVitisで高位合成中の最適化をやってもかまわない

Kria KV260 Vision AI Starter Kitを利用

- Kria KV26 SOM (System-On-Module)の一種
- ハードウェアの詳細は隠蔽されている
- Zynq Ultrascale+ を利用
 - しかし、機種もピン配置もオープンされていない
- KV260はVision application開発用
- ZynqをホストとしてVitis-AIを利用

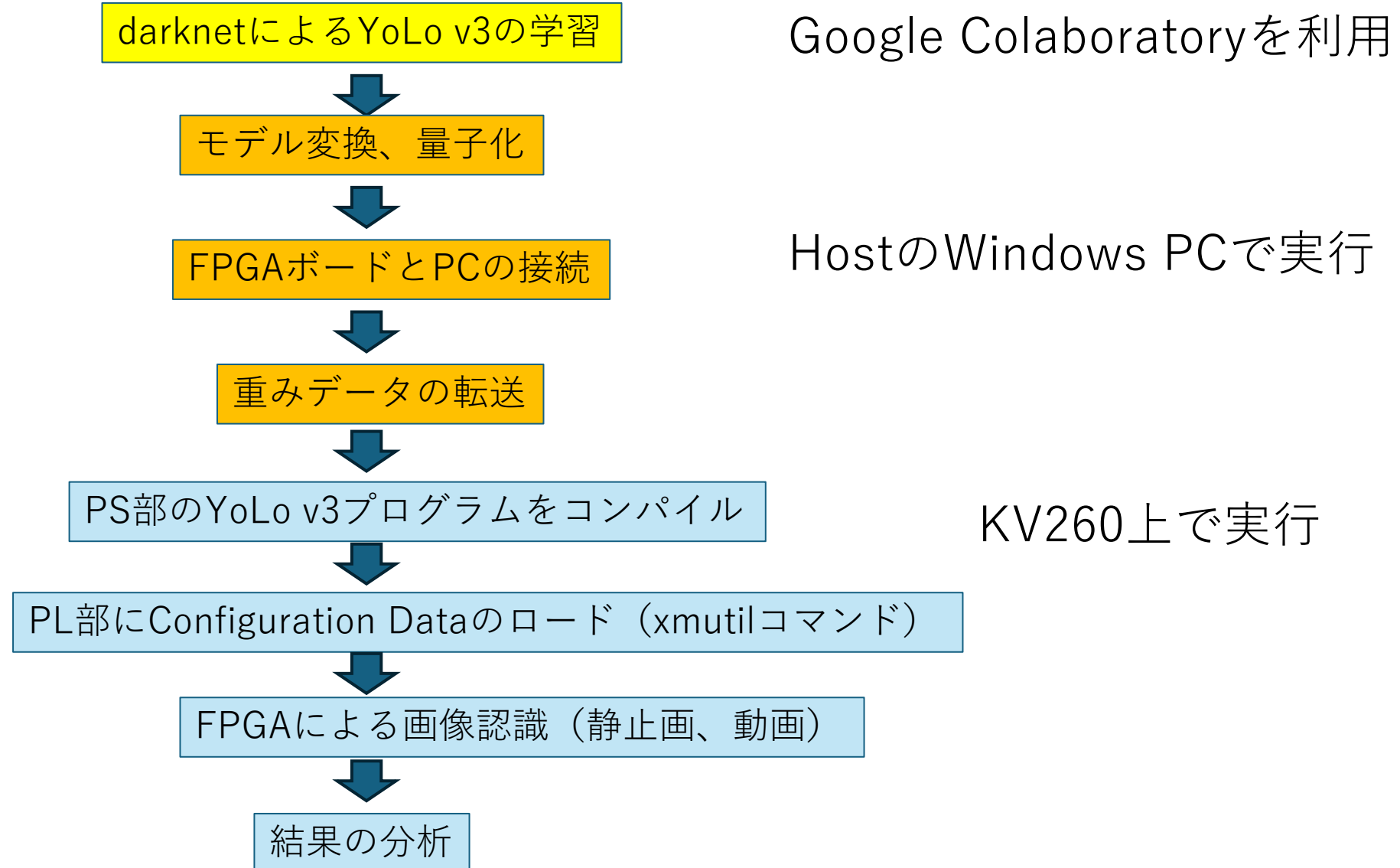


KV260ボードの内部構成

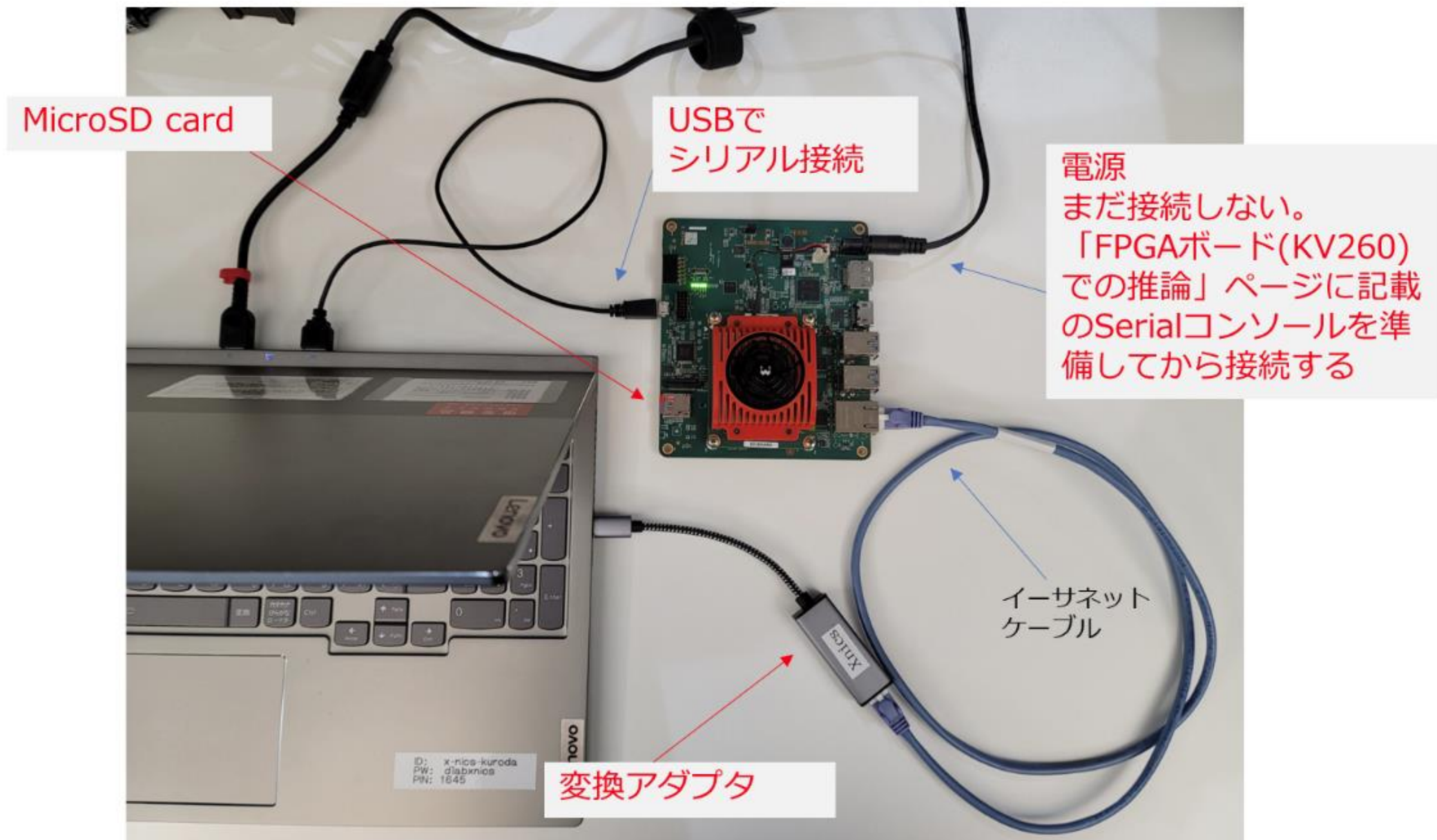


Yolo v3の実行 まずここまでやってもらう

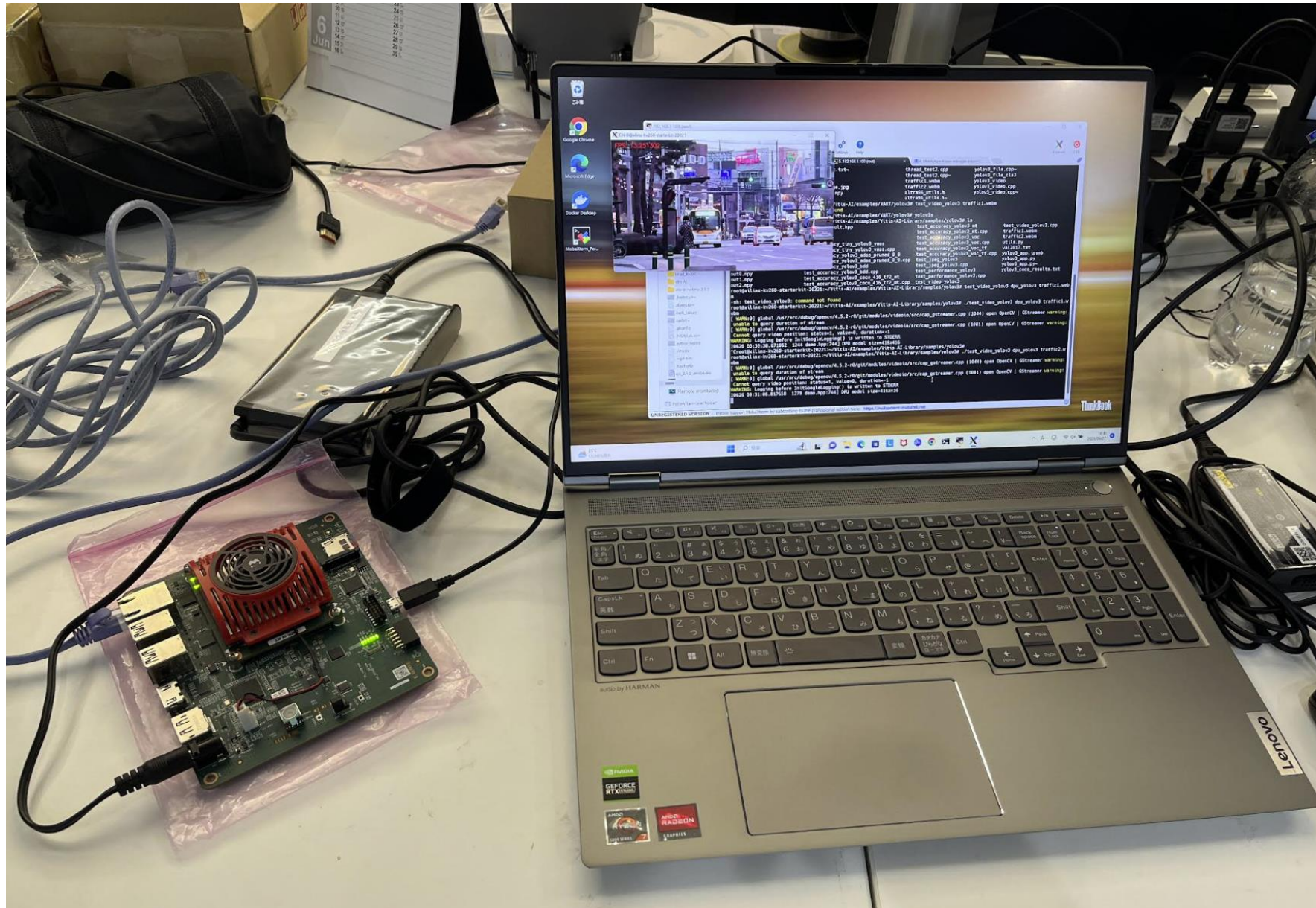
<https://github.com/takgto/utokyo-chipathon2023/wiki/>



KV260 の接続



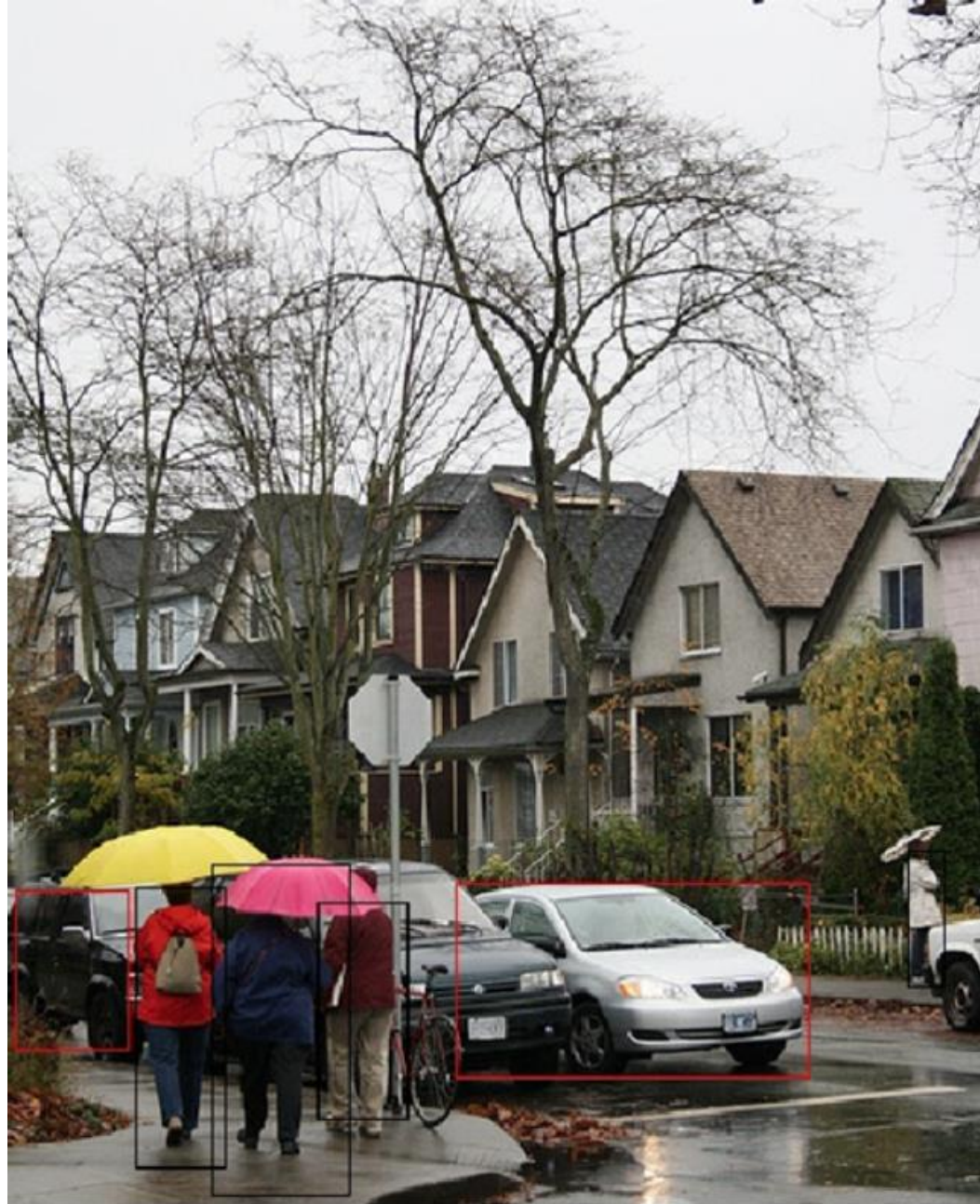
ホストPCと接続した様子



画像が出るまでの問題点

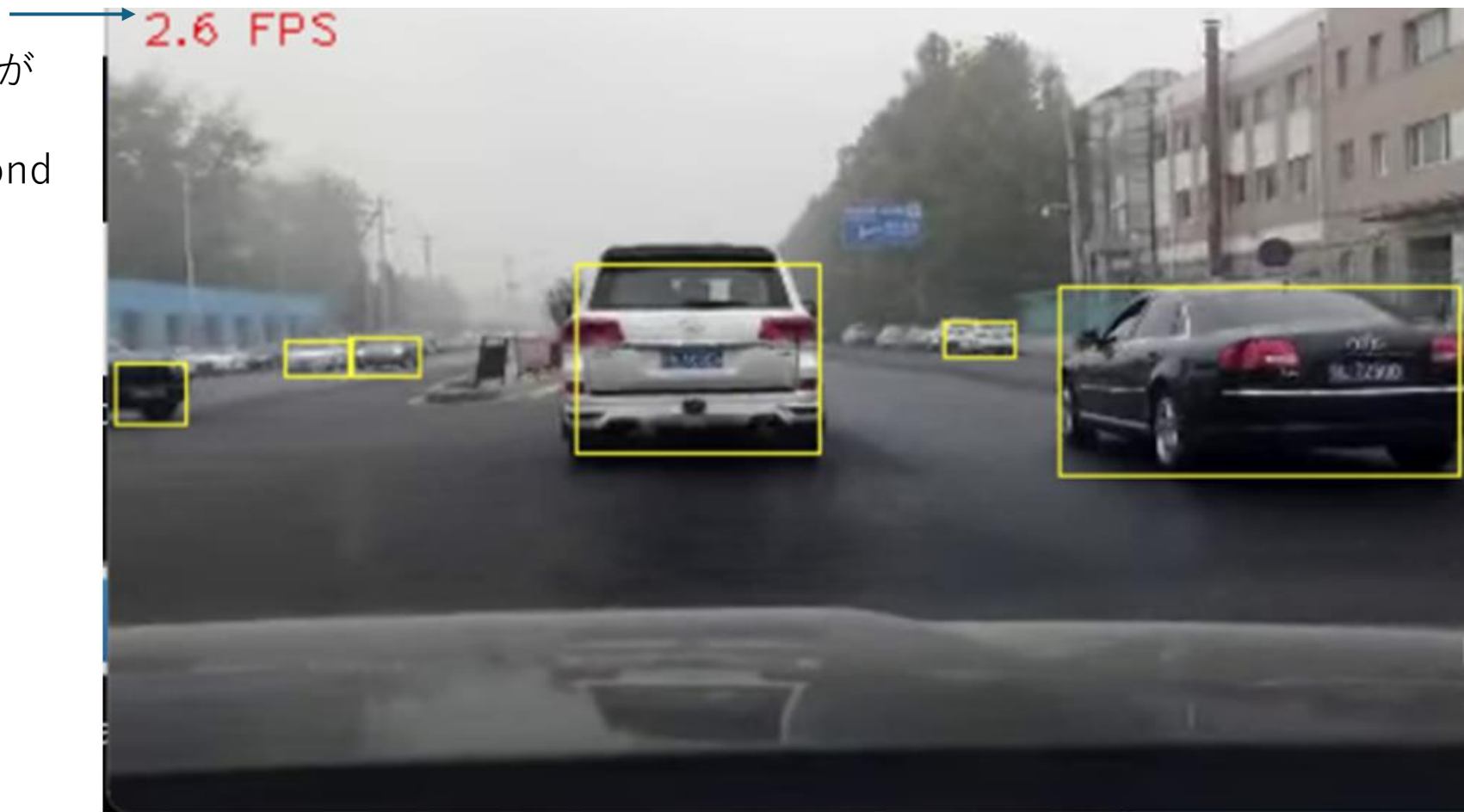
- google colaboratoryは時間制約が厳しいので十分学習ができない
 - 最初は学習済のを配布し、最適化時はGPU付のサーバーを使ってもらう
- WSL2 Linux環境→Docker起動→Conda起動
- FPGAボードをMobaxtermでシリアルで接続→Etherでローカル接続→VS codeでssh
 - 様々な環境を使いまくるため、学生は混乱する
 - Wikiを見ながらやると案外皆ちゃんと画像が出る
 - 東大生が優秀？ AIに興味を持っている人は、皆、結構この辺のツールは知っているのかも、
 - Webから色々ツールを取ってきてインストールしてなんとか動かすという作業には耐性があるようだ

静止画の表示



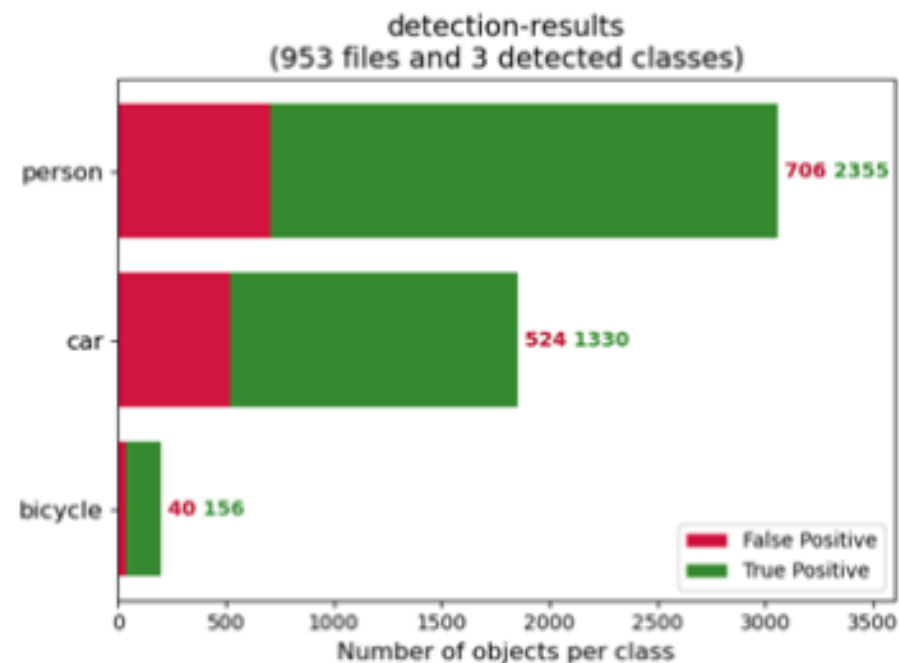
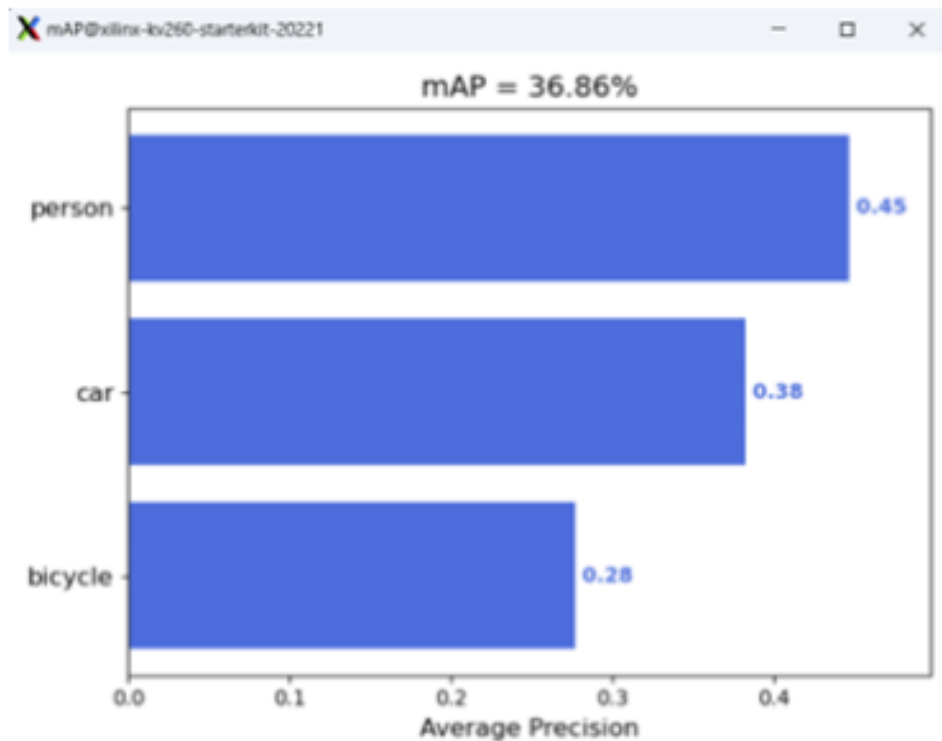
動画の表示

ここに性能指標が
表示される
Frame Per Second



精度の測定 mAP (mean Average Precision)

```
cd ~/Vitis-AI/examples/Vitis-AI-Library/samples/yolov3
./test_file_yolov3 dpu_yolov3-608 val2017.txt
cd mAP
python3 main.py -na
```



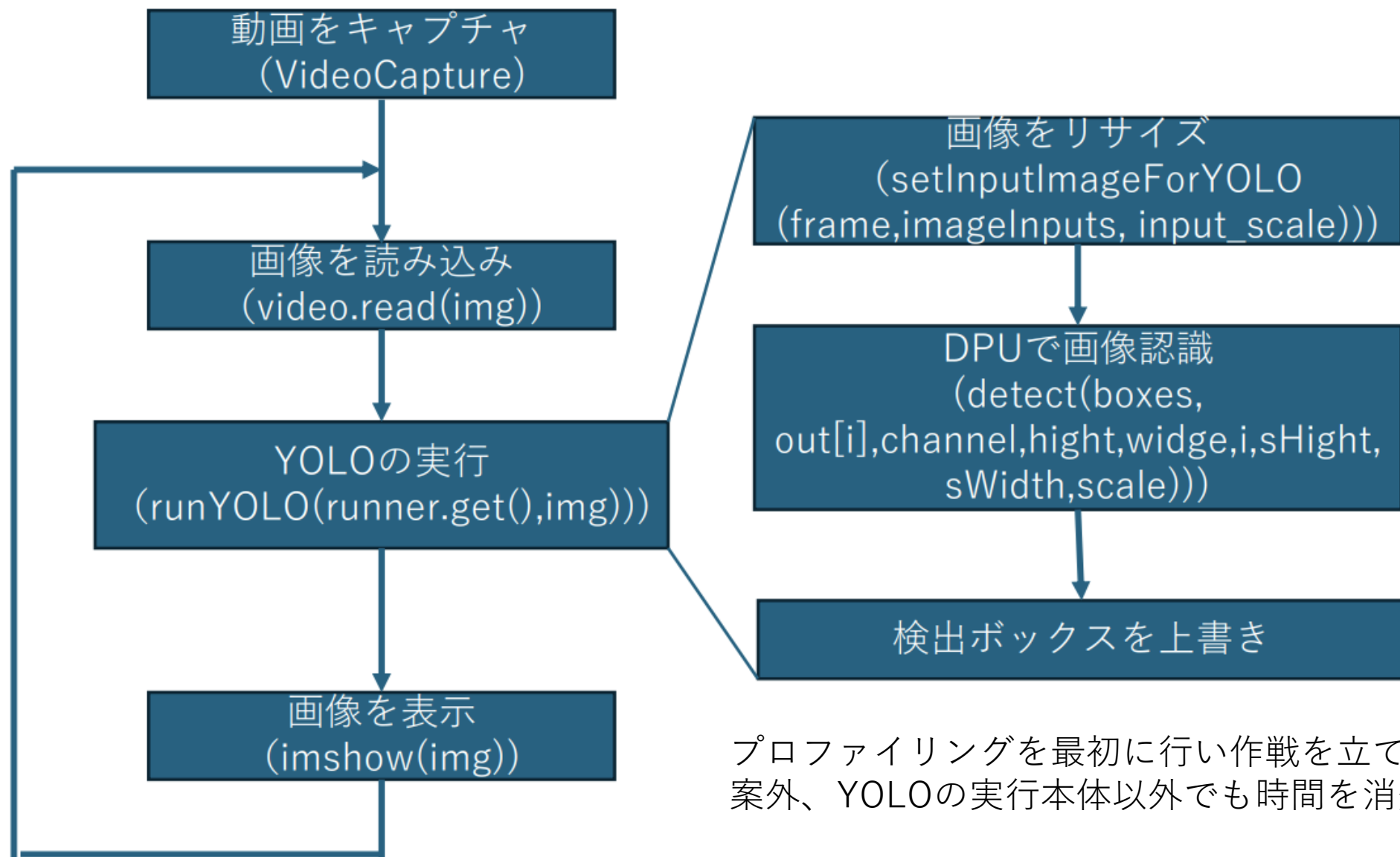
いよいよ最適化

- yolov3-tiny を実装する : モデルの作り直しが必要
- パイプライン化、マルチスレッド化: マルチスレッド化のみ行ったサンプルコードを示している
- 入力画像サイズを変えてみる : 精度は犠牲になるが性能は上げられる
- keras モデルを Onnx モデルに変えてみる
- Onnx モデルを使ってモデルを分割する
- DPU のチューニングを行う

これらのアイディア集はWikiに置いてある

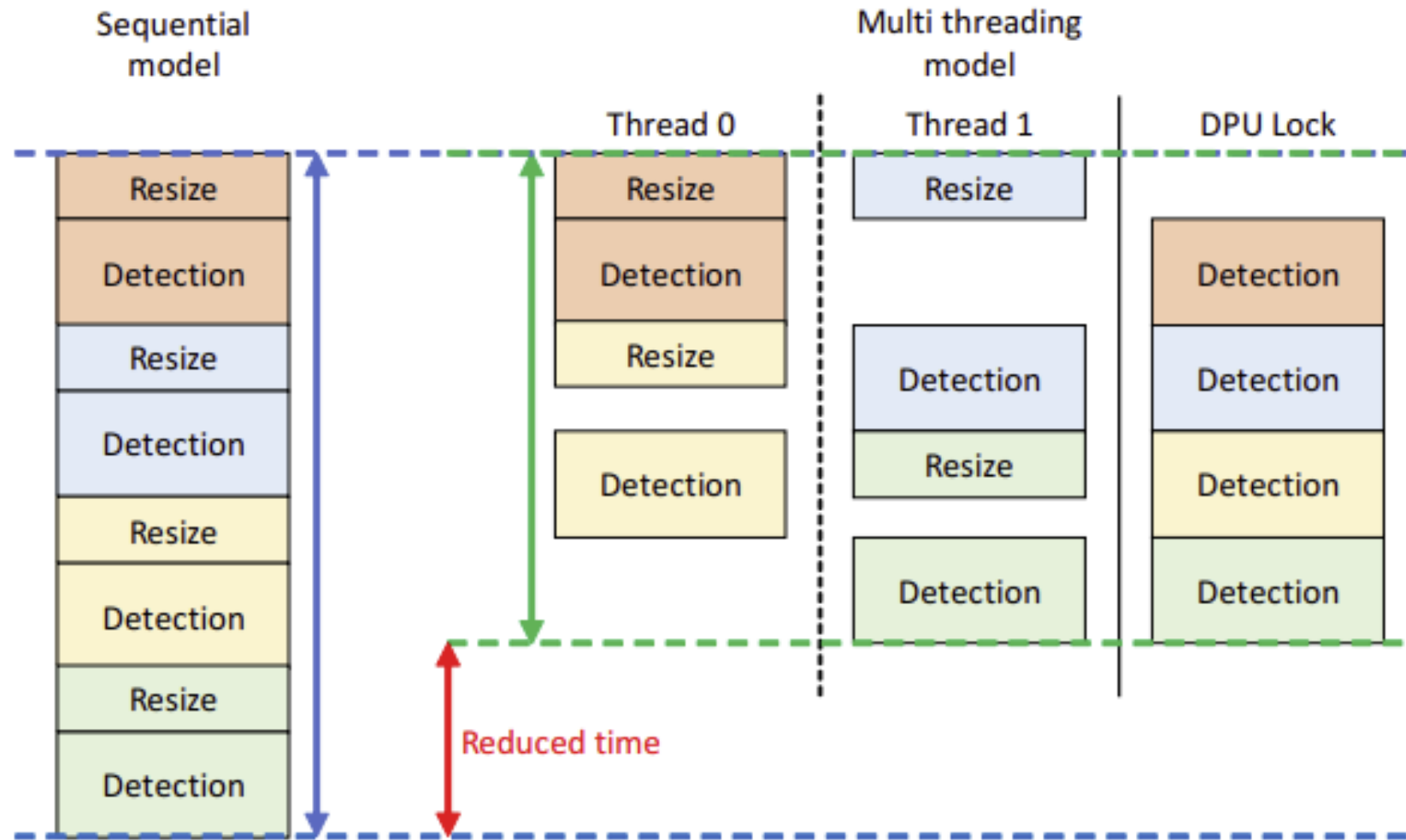
→<https://github.com/takgto/utokyo-chipathon2023/wiki/>

核となるプログラムの実行



プロファイリングを最初に行い作戦を立ててもらおう
案外、YOLOの実行本体以外でも時間を消費している

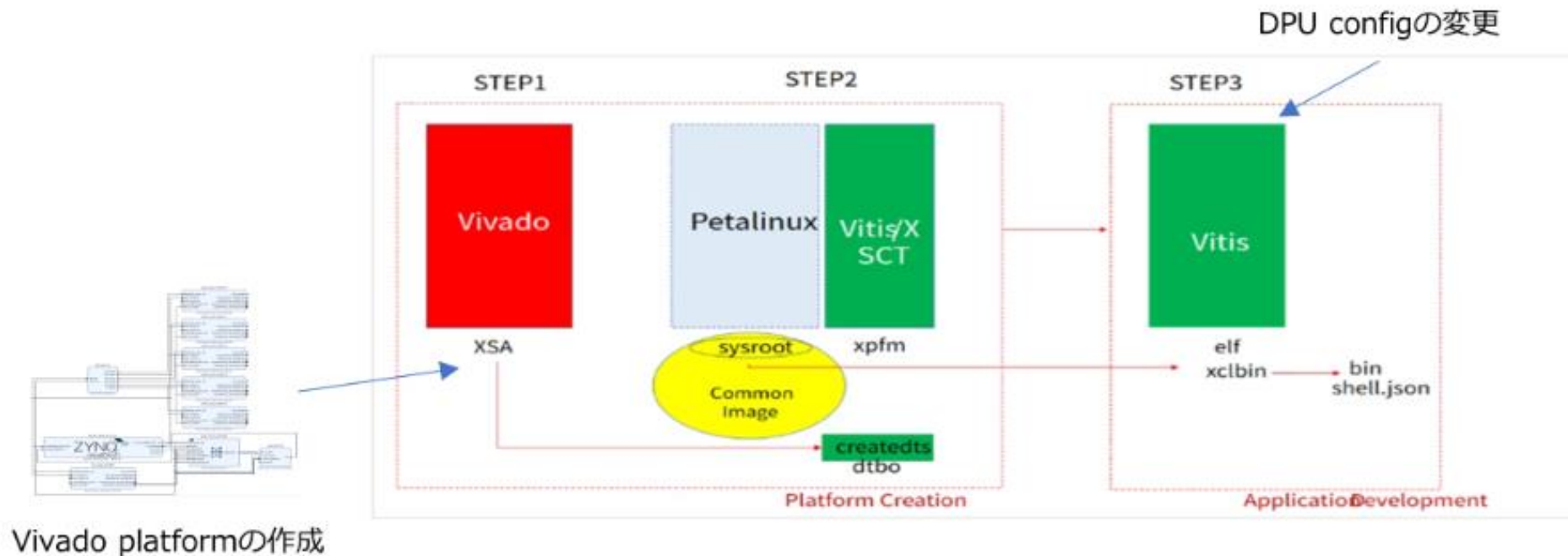
並列化、マルチスレッド化の例



これは関連研究①のAIエッジコンテストにおける一例

DPUを複数にする

- 基本設計ではDPUが1個だが、小規模なDPUを複数装備して並列実行する



これは、Vitis,Vivadoに関する知識が必要で、試みた人はほとんどいなかった
→潘君はうまく行った。スケジュールも工夫しないと早くならない

2024年東大でのハッカソン

- 1ターン4日、12人×4回 10:00-17:00 夏休み期間
 - 1日目：午前中は小菅講師の講義、Yolov3を動かすところまでやり最適化をスタートする程度
 - 2日目-3日目：最適化、研究室見学でチップを見せたりする
 - 4日目：午前でまとめて、午後は簡単なプレゼン
 - 4週間まとめて全体の発表会と日立中研での技術交流会を行う
 - サポートは小菅、後藤、新田見、天野の4名



東大目白台インターナショナルビレッジ
ワーキングコモンズ

2024年の試行（計47名、理系以外が3名）

学部生	20名	大学院生	27名
工・機械情報	6名	工・電気系	16名
工・電気電子/電子情報	3名	工・システム創成	2名
工・情報工学	2名	工・マテリアル	1名
工・精密工学	2名	工・先端学際	1名
工・システム創成	2名	工・精密工学	1名
工・マテリアル	1名	工・社会基盤学	1名
工・航空宇宙	1名	総合文化・広域科学	1名
教養・教養学科	2名	新領域	2名
文・人文	1名	情報理工・電子情報	1名
		情報理工・知能機械	1名

設計例

	FPS	mAP (%)		最適化手法
最適化前	2.6	45	YoLo v3	何もやってない
御手洗	75.0	24.3	YoLo v3 tiny	display/Frame関数の起動タイミング調整 waitKeyの頻度を減らす 並列実行でのスレッドの順序制御 画像サイズの変更
潘	9 101	47 12	YoLo v3 YoLo v3 tiny	複数スレッドでDPUの複数利用 YoLo v10-nなど新しいモデルも試す Resize, 量子化に合わせた再学習
于	70.3	25.1	YoLo v3 tiny	スレッド数の調整 表示頻度の削減

おわりに

- ハードウェア設計技術を知らずにAIと画像処理から最適化を行う
ハッカソンを提案
- 次回の研究会で一般試行を企画中
 - KV260とアクセサリを貸与、3名までグループ可、指導教員の保証が必要
 - KV260は安いが国家予算で買ったので、貸与したきりになると凄くまずいので、
 - ソフトウェア、標準デザインはすでにGitHub上にある
 - Vivadoなどの設計ツールが使えなくても参加可能
 - Windows PCにWSL2、docker, conda, Mobaxtermがないと標準デザインが使えない
 - もちろんMacで相当のを用意すれば使えるがトラブルがあるだろう
 - 3週間程度の設計期間を設ける
 - YoLo v3が動くまでは天野がサポートするが、そこから先は自分でやる
 - 本番は違った動画でFPSとmAPを測定して順位を決める
 - 本番後KV260を回収
 - 賞金が出るかは怪しい（試行なんでスポンサー集めがこれから、賞状は出る）
- うまく行ったら同一コンセプトでLLMを実装するのをやりたいが、
現在苦戦中