DPUを用いたマルチタスクDNN表情認識システムのFPGA実装

Multi-task DNN Facial Expression Recognition System using DPU on FPGA

電気電子情報工学専攻　AES2301　安藤 拓翔(指導教員　井上 優良)

Key Words: DNN, FPGA, DPU, Facial Expression Recognition

1. 緒言

人物の表情は，人間のコミュニケーションにおいて感情や意図を伝達するための有効な非言語手段[1]である．同時に，人間とコンピュータ間のコミュニケーションに対しても有効であり，ペット型ロボットや医療用ロボット[2],[3]に応用されている．これまでは，サポートベクターマシンのような機械学習を用いた手法[4]が提案されてきた．近年では，DNN(Deep Neural Network)ベースの手法[5],[6]が主流となっており，従来の手法より大幅な認識精度の向上を達成している．画像処理による表情認識は，前処理として顔の関心領域を抽出し，関心領域内で表情認識を行う．従来の顔検出では，Viola-Jones Haar Cascade分類アルゴリズム[7]が用いられてきた．現在では，この顔検出もDNNを用いた手法が主流となっており，R-CNN[8]やYOLO[9]による高精度な検出手法が提案されている．このように，表情認識システムを構成する顔検出と表情認識はDNNを用いた手法により，実応用可能な精度の達成を見込める．

一般的にDNNの推論には，GPUが利用され，高い演算性能を発揮する．一方で推論時に膨大な消費電力を要求するため，バッテリー駆動の制約があるロボットには不向きである．そのため，GPGPUの代わりとして，FPGA(Field Programmable Gate Array)による実行が注目されている．FPGAは，DNNによる推論処理を高速化できる性能を持ち，低消費電力な演算が可能なデバイスであり，ロボットのような組込みシステムでのDNN推論実行に適している．

本稿の残りの部分は以下のように構成されている．第2節では先行研究を紹介し，本研究の位置付けを示す．第3節では，DPUによる表情認識システムについて説明する．第4節では本システムの評価と考察を行う．最後に，第5節では我々の研究を結論づける．

2. 関連研究

FPGAを用いて表情認識を実装した研究はいくつか報告されている．そのほとんどが顔検出によって適切に顔画像を取得可能であることを前提としたアプローチであるため，表情認識の手法のみ提案されている[10],[11]．それに対してVinhらは，SoC FPGAで顔検出を実行したのち，検出した顔に対して表情認識を実行するスタンドアローン型の表情認識システムを実装した[12]．この表情認識システムでは，顔検出はARMのCPUで実行され，表情認識はFPGAで実行される．表情認識はDNNによる推論がFPGAで実行されるため，高い精度を達成した．その一方で，顔検出ではFPGAリソースの制約のため，OpenCVのHaar Cascade検出器が用いられている．このHaar Cascade検出器は演算性能の低い組込み用CPUでも実行可能であるが，カメラに対して正面の顔のみ検出可能で，顔が斜めや横向きになる場合の検出精度が低い．さらに，照明条件が一定でない場合にも性能が低下する課題を持つ．ゆえに，非DNNによる顔検出では現実世界で適切な検出ができず，表情の認識精度が低下する．

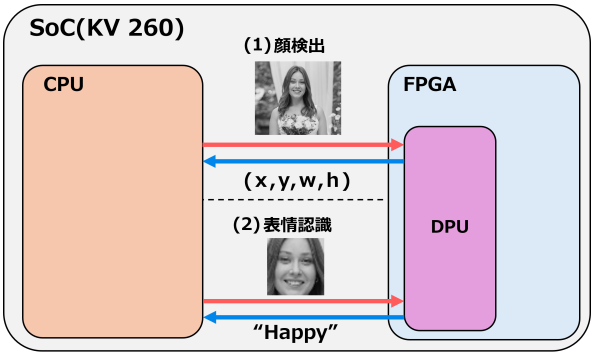

図 1: 提案手法のハードウェア構成

先行研究のハードウェア構成では，顔検出をDNN推論で実行するには，新たに顔検出専用アクセラレータを加える必要がある．これは，回路規模の大幅な増大を意味しており，FPGAリソースの制約から困難である．そこで本研究では，汎用CNNアクセラレータであるDPUを利用して，顔検出もDNN推論で実行可能なシステムを提案する．DPUはXilinxが提供する汎用的なCNNアクセラレータであり，異なるCNNの推論処理を同一のDPUで実行できる．このDPUを用いることで回路規模の増大を抑えつつ，表情認識システムを実装した．

3. FPGAによる表情認識システム

3.1. システムの概要　本システムの処理は2つのステップに分けられ，図1に示すように(1)顔検出ステップと(2)表情認識ステップで構成されてい

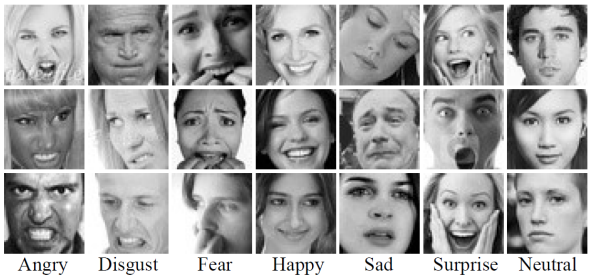

図 2: FER-2013 dataset のサンプル画像

る．顔検出ステップでは，カメラのフレームを入力として，DNNモデルであるDenseBoxによる顔検出が行われる．次に，表情認識ステップでは顔の関心領域を対象にCNNの表情認識モデルにより表情を識別する．これらのステップで行われる推論処理は，それぞれFPGA部で実行される同一のDPUにオフロードされる．また，全体の処理のうち，画像のリサイズ，トリミング，グレースケール化などの前処理はCPUで実行する．

では，顔検出と表情認識のDNNについて説明する．顔検出モデルはXilinxが提供しているDenseBoxモデル[13]を利用した．DenseBoxによる推論は非常に軽量なため，演算性能の低いDPUで実行した場合でも高速な検出が可能である．このモデルの学習にはWider faceデータセット[14]が用いられている．

一方で，表情認識モデルは，Guarnizらが作成した表情認識モデルのネットワーク[15]を参考にして作成した．入力された48×48 pxのグレースケール画像から，7つの異なる表情クラス(Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral)を識別する．このモデルの学習には，FER-2013[16]を利用した．図2のように，このデータセットは合計35,887枚の48×48 pxのグレースケール画像で構成され，7つの表情ラベルが付与されている．

3.2. FPGA SoC アーキテクチャ　本システムは，ARM Cortex-A53プロセッサとXilinxのFPGAであるUltraScale+ を統合したアーキテクチャを持つXilinx Zynq Ultrascale+ MPSoCに実装される．評価用のボードとして，このSoCを搭載した開発ボードであるXilinx Kria KV260を利用した．また，DPUを組み込んだハードウェアデザインの作成にはXilinx Vivado 2021.1を用いた．DPUのハードウェアアーキテクチャを図3に示す．DPUは，このFPGA部に組み込まれており，CPUとAXI(Advanced eXtensible Interface)バスで接続されている．CPUとDPUは，オフチップメモリを共有しており，DPUはこのオフチップメモリから命令をフェッチして演算エンジンの動作を制御する．一方でFPGA上のオンチップメモリには，入力画像やNNパラメータ(重みやバイアス)や中間特徴量のバッファとして使用することで、高スループット

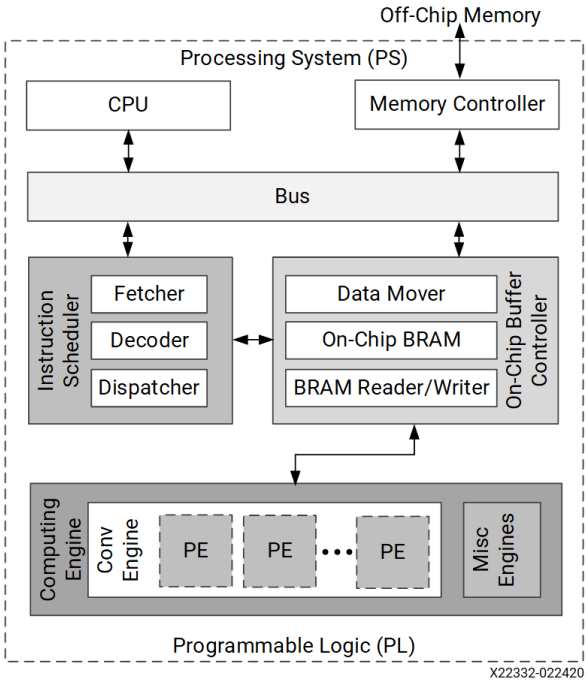

図 3: ハードウェアアーキテクチャ[17]

トと高い効率性を実現している．PE(Processing Element)は，DPUの演算エンジンであり，多段パイプラインで動作することで高い演算性能を実現している．

3.3. DPUへのオフロード　DPUは演算性能の異なる複数のアーキテクチャをサポートしており，用途別にこれを選択して利用することできる．DPUのサイズの大きさ畳み込みユニットの並列度に依存しており，B512からB4096までの異なるサイズが提供されている．512や4096は，1サイクルあたりの演算実行回数を示す．演算性能の高いDPUを回路に組み込む場合は，FPGAリソースの消費量が増加するという課題がある．本システムで実行される顔検出や表情認識は，3D物体検出のような他のDNN推論と比較して軽量な処理である．そこで，本システムは最も回路規模の小さいB512を1つFPGA部に実装した．

先行研究で実装されたアクセラレータは，高位合成により表情認識のDNNモデルをそのまま回路化する専用回路型である．そのため，表情認識のDNN推論のみ実行可能であり，顔検出は別のアクセラレータを実装する必要がある．一方で．本システムで利用したDPUはシストリックアレイ型であり，異なる種類の推論を同一のアクセラレータで実行可能である．本研究では，DPUで顔検出と表情認識を時分割で実行することで，アクセラレータを増やさずにFPGAリソースを効率よく利用する手法提案する．

DNNをDPUにオフロードするには，Vitis AI 2.5のツールを用いたモデルの変換を行う必要がある．Vitis AIはXilinxが提供するDNNフレームワーク

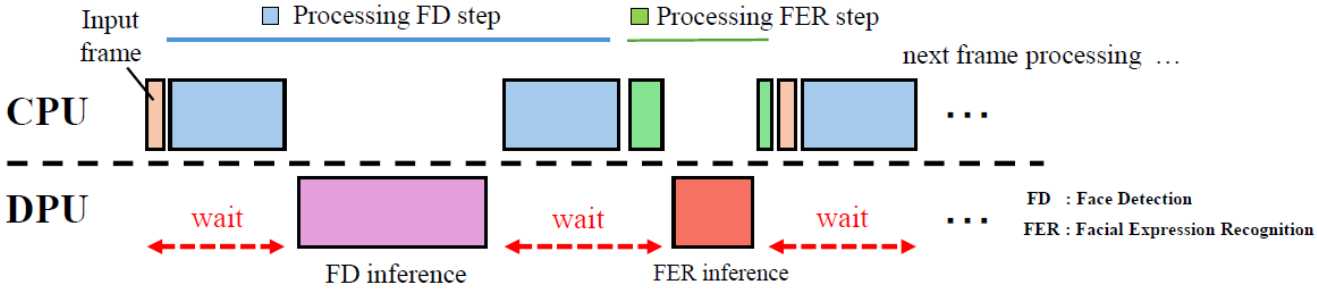

図 4: シングルスレッドによる顔検出と表情認識の処理フロー

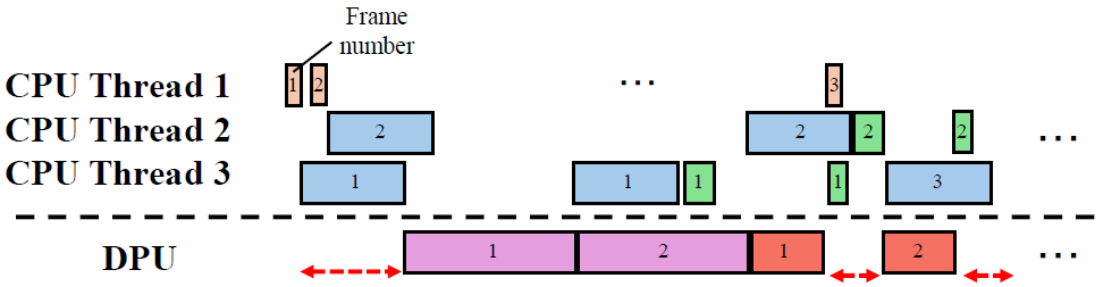

図 5: マルチスレッドによる顔検出と表情認識の処理フロー

で，構築したモデルをDPUで実行可能な形式に変換することが可能な開発環境である．DNNモデルの変換には、このVitis AIのツールに含まれるAI quantizerとAI compilerを使用した．AI quantizerは32 bitの浮動小数点型のモデルを8 bitの固定整数型のモデルに量子化を行うツールであり，可能な限り精度を維持する．このように，モデルの量子化を行うことでメモリの使用量を最適化でき，ハードウェアの演算回数を削減することができる．一方で，AI compilerはAI quantizerによって8 bitに量子化されたモデルをDPU上で実行可能なモデルに変換するツールのことである．これらのツールにより，モデルの量子化とコンパイルを行うことで，DNNモデルをDPUで実行することが可能になる．

3.4. マルチスレッド戦略　本システムを1スレッドで実行した場合の処理フローを図 4に示す．このようにCPUが処理をしている間，DPUは次の処理を待機しており，効率的な利用ができていない．そこで，本システムではマルチスレッドを活用することで，DPUの利用効率を向上させる．本システムはPythonのマルチスレッドライブラリであるThreadingを用いることで顔検出と表情認識を異なるスレッドで実行する．図 5にマルチスレッドで実行した場合の処理フローを示す．マルチスレッドの前提として，OpenCVによるカメラ入力がスレッドセーフではないため，カメラ入力専用のスレッドを1つ用意する必要がある．そのため，カメラ入力専用スレッドと2つの処理スレッドをたて，合計3スレッドで実行する．2つの処理スレッドで異なるフレームの処理を実行し，これらのスレッドは1つのDPUを共有する．この実装方法は1スレッドによる実行と比較して，DPUの利用効率が向上し，全体のスループットが向上することが期待される．本システムでは，このCPU処理のマルチスレッド化によるDPUの実行を提案し，その有効性を評価する．

4. 実験と考察

本節では，実験方法と結果を説明し，本システムの性能を評価する．

4.1. 顔検出　本実験の目的は，先行研究との比較によりDPUで実行される顔検出モデルを評価することである．認識精度(Average Precision)とレイテンシの観点から評価を行う．ここでは，DPUによる顔検出の性能を，AFWデータセット[18]を評価用データセットとして，本手法を先行研究と比較する．AFWデータセットは205枚のさまざまな解像度の画像で構成されており，で473個の顔の注釈が付与されている．画像は特に横からの角度のある顔やスケール、照明、オクルージョンなどの課題を持った画像が含まれている．

AFWデータセットを用いて，顔検出における評価を行った結果を表 1に示す．DPUで実行されるモデルと，先行研究で用いられたHaar Cascade検出器の比較を行う．また，DenseBoxの入力サイズは640×480 pxであり入力画像をリサイズする必要がある．一方で，Haar Cascade検出器はリサイズを行わずに推論を行う．そこで，DenseBoxのレイテンシにリサイズの時間も加えることで公平な比較を行う．

DPUで実行した量子化モデルの精度は0.917となり，先行研究のHaar Cascade検出器による手法

表 1: 顔検出の精度とレイテンシの結果

Method		Average Precision	Latency[ms]
CPU	Haar Cascade (Previous work)	0.531	798
DPU	**DenseBox (Our work)**	**0.917**	**42.10**

表 2: 表情認識の認識精度とレイテンシ

Method	Accuracy[%]	Latency[ms]
CNN(Previous work)	66	6.36
CNN(Our wor)	**67.4**	**7.34**

と比較して，精度が約1.73倍向上した．レイテンシはDPUで実行したモデルが42.10 msであり，先行研究と比較して約18.95倍短縮することが確認できた．したがって，認識精度とレイテンシの観点でDPUによるDNNモデルの顔検出は，先行研究より優れることが確認できた．

4.2. 表情認識　本実験の目的は，先行研究との比較によりDPUで実行される表情認識モデルを評価することである．認識精度(percentage)とレイテンシの観点から評価を行う．評価用のデータセットは，FER-2013[16]を利用した．それぞれのDNNによる表情認識の精度とレイテンシを表 2に示す．

DPUで実行した表情認識モデルの精度は67.4％であり，レイテンシは7.34 msであった．先行研究での，FPGAにオフロードが行われた表情認識モデルは，FER-2013のテストデータセットに対して精度が66％であり，レイテンシが6.36 msという性能であった．表情認識の精度においては，本システムの方が優れるという結果になったが，レイテンシは先行研究の方が短いことが分かった．

4.3. システム全体の比較　B512を用いた本システムのハードウェア構成の有効性を検証する．本システムの構成を先行研究のシステム構成と異なるサイズのDPUを組み込んだシステム構成に対してFPGAリソース消費量，システムの消費電力とスループットを指標として評価を行う．さらに，マルチスレッドによるDPUの利用効率と処理性能の変化についても評価を行う．表 3に，これらを比較した結果を示す．ただし，先行研究の実装環境を完全に再現することはできないため，ハードウェア構成のみを再現した．具体的には，CPUでのHaar Cascade検出器による顔検出と，CNNによる表情認識をDPUで実行する構成のシステムを作成して比較を行った．消費電力については，アイドル状態(7.8[W])とシステムが実行されている状態のボード全体の最大消費電力の差を計測し，システムの消費電力を算出した．スループットは，顔検出と表情認識を含むシステム全体のスループットを計測した．

まず，FPGAリソース消費量とスループットについて評価する．先行研究はALMを22,465個，DSPを112個消費していた．それに対して本システムではLUTを27,023個，DSPを118個消費していた．回路規模を比較すると，先行研究の方がFPGAリソースの消費量が少ないことが分かった．本システムは先行研究と比較して回路規模は大きいが，同一のDPUで表情認識と顔検出の推論の実行を行うことが可能である．先行研究のスループットが11.67FPSであるのに対し，本システムは1スレッドで14.69FPS，2スレッドで25.00FPSを達成している．リアルタイム性を有する顔検出と表情認識のシステムとして，本システムは十分なスループットを達成した．

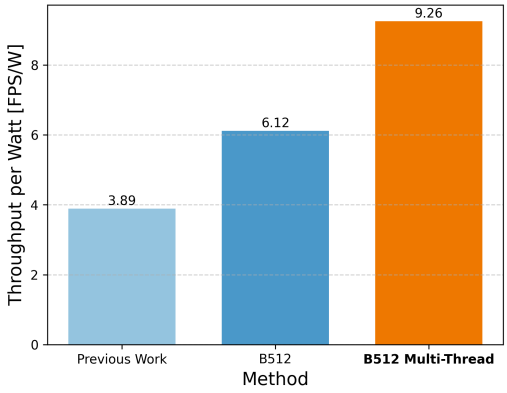

図 6: 消費電力当たりのスループットの比較

次に，消費電力とスループットについて評価する．先行研究の消費電力は2.3[W]であるのに対し，本システムは1スレッドで2.4[W]，2スレッドで2.7[W]であった．先行研究と比較して，本システムは消費電力が増加しているが，スループットは2スレッドで約2.14倍向上している．図 6に消費電力あたりのスループットを示す．B512を用いた本システムのスループットあたりの消費電力は，1スレッドで6.12 FPS/W，2スレッドで9.26 FPS/Wであった．マルチスレッドによる本システムは，先行研究と比較すると，約2.4倍向上しており，消費電力とスループットの観点においても本システムの有効性が確認できた．

では，DPUの利用効率について評価する．表 3のDPUの利用率は，システムの総実行時間に対するDPUの実行時間の割合を示している．CPUとDPUは並列に実行できるため，この利用率が高いほどDPUが効率的に利用されている．1スレッドでは顔検出と表情認識の合計が22.85%であり，2スレッドでは78.44%であった．このように，マルチスレッドによるDPUの利用効率は高く，システム全体のスループット向上に寄与していることが分かった．

5. 結言

本研究では，汎用CNNアクセラレータである

表 3: 先行研究のシステム構成に対する性能の比較

Method	FPGA resource			Thread	Throughput [FPS]	Peak Power [W]	Power [W]	DPU utilization [%]		
	ALMs or LUTs *	DSPs	BRAMs					FD	FER	Total
Haar Cascade (Previous work)	34,593	230	44	1	11.67	10.1	2.3	-	-	-
Our work	**27,023**	**118**	**12**	1	14.69	10.2	2.4	15.84	7.01	22.85
				2	**25.00**	**10.5**	**2.7**	**49.72**	**28.72**	**78.44**

(*先行研究はintelのボードのためALMで構成される)

DPUによる表情認識システムを実装した．シストリックアレイ型のアクセラレータを利用することで，顔検出と表情認識を行う2つのDNNの推論を時分割で同一のDPUで実行することが可能である．また，DPUの利用効率を上げつつ，全体のスループットを向上させるために，マルチスレッドを活用したシステムを提案した．DenseBoxによる顔検出は従来のHaar Cascade検出器による手法と比較して，精度が約1.73倍向上し，レイテンシは約18.95倍短縮することに成功した．一方，表情認識の精度は67.4％を達成し，画像１枚当たりに要するレイテンシは 7.34 ms であった．

また，先行研究と比較して回路規模はやや大きいが，同一のDPUで表情認識と顔検出の推論の実行を行うことが可能である．さらに，マルチスレッドによる実行では，先行研究より消費電力あたりのスループットは約2.4倍向上し，25FPSのスループットを達成した．したがって，同一のDPUによるマルチスレッドを活用したハードウェア構成は回路規模を抑えつつ，先行研究よりも良好な結果を達成することができたと結論付ける

今後の課題としては，十分なスループットを維持しつつ，さらなる低消費電力化が挙げられる．例えば，顔の位置が変わらないため顔検出を数フレームに１回にするなど，実際の運用に合わせた最適な処理を検討する．

参考文献

(1) Y.-I. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 2, pp. 97–115, 2001.

(2) X. Li, C. Lou, J. Zhao, H. Wei, and H. Zhao. "tom" pet robot applied to urban autism. *ArXiv*, Vol. abs/1905.05652, , 2019.

(3) O. Arriaga, M. Valdenegro-Toro, and P. Plöger. Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*, 2017.

(4) D. Ghimire, S. Jeong, J. Lee, and S. H. Park. Facial expression recognition based on local region specific features and support vector machines. *Multimedia Tools and Applications*, Vol. 76, pp. 7803–7821, 2017.

(5) M. Rusia and D. Singh. An efficient cnn approach for facial expression recognition with some measures of overfitting. *International Journal of Information Technology*, Vol. 13, , 09 2021.

(6) J. Shao and Y. Qian. Three convolutional neural network models for facial expression recognition in the wild. *Neurocomputing*, Vol. 355, pp. 82–92, 2019.

(7) P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Conf Comput Vis Pattern Recognit*, Vol. 1, pp. I–511, 02 2001.

(8) X. Sun, P. Wu, and S. C. Hoi. Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing*, Vol. 299, pp. 42–50, 2018.

(9) W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang. Yolo-face: a real-time face detector. *Vis. Comput.*, Vol. 37, No. 4, p. 805âĂŞ813, apr 2021.

(10) J. Kim, J.-K. Kang, and Y. Kim. A resource efficient integer-arithmetic-only FPGA-based CNN accelerator for real-time facial emotion recognition. *IEEE Access*, Vol. 9, pp. 104367–104381, 2021.

(11) H. Phan-Xuan, T. Le-Tien, and S. Nguyen-Tan. FPGA platform applied for facial expression recognition system using convolutional neural networks. *Procedia computer science*, Vol. 151, pp. 651–658, 2019.

(12) P. T. Vinh and T. Q. Vinh. Facial expression recognition system on SoC FPGA. In *2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pp. 1–4. IEEE, 2019.

(13) A. Xilinx. Vitis ai library user guide g1354 (v3.5) june 29. `https://docs.amd.com/viewer/book-attachment/KR753m2y6vGxH3r37gIq3A/cBdlK8Cf7joC9NuWU_MHjg`, 2023. (Accessed on 04/07/2024).

(14) S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5525–5533, 2016.

(15) Facial-expression-recognition-2018. `https://github.com/kckeiks/Facial-Expression-Recognition-2018/tree/master`. (Accessed on 09/07/2023).

(16) I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio. Challenges in representation learning: A report on three machine learning contests. arXiv 1307.0414 stat.ML, 2013.

(17) Xilinx and inc. dpuczdx8g for zynq ultrascale+ mpsocs product guide (pg338),2023. `https://docs.amd.com/r/en-US/pg338-dpu/Introduction?tocId=3xsG16y_QFTWvAJKHbisEw`. (Accessed on 07/12/2024).

(18) X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2879–2886, 2012.