

MRHMD を用いたコンクリート壁面の インタラクティブひび割れ検査システムの研究

A Study on Interactive Crack Inspection System on Concrete Wall Using MRHMD

TAEJOON PARK

(指導教員 山口 友之)

Abstract – Automated crack inspection on the concrete structure is an important task to evaluate the necessity for its repair. Machine Learning (ML) based image processing technology has been proposed as an automated crack inspection method. However, the accuracy is still inferior to human work. Besides, inspecting fine cracks, which is hard to find out by humans at a far distance, is an important task to estimate the factors of deterioration of concrete structures such as concrete carbonation, corrosion of steel, etc. In this research, we proposed ML based automated crack inspection and visualization method, which human inspector can adjust undetected or erroneous detection of the inspection result, which can be used as ML training data. As a crack detection method, we used Encoder-Decoder based Convolutional Network (CNN), which is capable of pixel-level segmentation. As a crack visualization method, we used Mixed Reality Head-Mounted Display (MRHMD) to show the inspect results visually, which fine cracks can be easily recognized by human inspectors from far distance. The proposed method confirmed the improvement of the accuracy of crack inspection by presenting new learning data to the system. Furthermore, we confirmed that displaying cracks using MRHMD has proved effective in showing fine cracks that human is hard to see.

1 はじめに

日本の数多くのコンクリート構造物は高度経済成長期に建てられ、建築後 50 年以上を経過する構造物の数は年々増加している[1]。そのため、老朽化するコンクリート構造物の耐久性の低下を防ぐために定期的なメンテナンスが必要である。コンクリート構造物の老朽化より起こる現象の一つであるひび割れは、人には見えにくい 0.2mm 前後の微小なひび割れ幅でもコンクリートの中性化や鋼材の腐食に影響を及ぼし、劣化の原因となるため、ひび割れの初期発見が重要である[2]。このように、ひび割れを早期発見して、修繕の必要性を評価し、構造物の安全性を維持するひび割れ検査は重要な工程である。

従来のひび割れ検査で主に行われている方法としては、近接目視による記録方法がある。検査員が検査対象に接近し、クラックスケールなどを用いてひび割れの位置、長さ、幅を記録する方法である[3]。

しかし、日本では少子高齢化が進行し、熟練検査員が不足する課題が生じているため[1]、検査の自動化が強く求められている。

ひび割れ自動検査の研究として、画像処理を用いたひび割れ検出方法が数多く行われている。画像処理によるひび割れ検出は、検査対象に接近せずに、カメラで撮影した画像だけでひび割れを検出できる方法で[4]、既に実現場で適用され始めている。しかし、壁面の汚れや照明条件など、撮影環境によって結果の精度が多く異なる問題がある[4]。

その解決策として、Machine Learning (ML) の一種である Convolutional Neural Network (CNN) を用いたひび割れ検出の研究も行われている[5]。ML を用いたひび割れ検出はひび割れ検出画像の撮影条件に大きく影響されずに、ひび割れを効果的に検出できる。しかし、古典的な CNN によるひび割れ検出は、ひび割れ検出の結果がピクセル単位ではなく、ひび割れがある大まかな領域を区別することしかで

きない。ひび割れ領域をピクセル単位で検出するために提案されている Encoder-Decoder based CNN を用いた Semantic segmentation の手法は、学習過程でダウンサンプリングされる情報を用いて、学習結果をアップサンプリングすることで、ピクセル単位でのひび割れ検出を可能にする方法である[6]。しかし、ML を用いたひび割れ検出では、検出結果の誤検出を除去するためにプリプロセスやポストプロセスを用いる手法が提案されているが、これらの方法は、微小なひび割れまでノイズと判断し、除去される問題がある[7]。

これまでにひび割れ検出結果の提示手法として、Mixed Reality Head-Mounted Display (MRHMD) を用いて、壁面に検査結果をリアルタイムで提示する手法が提案されている[8]。このシステムは事前に作られたひび割れ画像を実空間上に投影して表示するシステムである。しかし、このシステムはひび割れ画像を事前に用意し、データベースを作成する必要があるため、壁面の材質の違いや、照明の有無など、実現場での多様な撮影環境に合わせたひび割れ検査はプロセスの実現が困難である。

したがって、ひび割れ検出結果をリアルタイムで提示し、インタラクティブに多様な撮影環境に合わせたひび割れ検査効率向上のための全体的なシステムに関する議論が必要である。

そこで、本研究では、Encoder-Decoder based CNN を用いた Semantic segmentation 手法によって検出されたひび割れ画像に対して、人が誤検出をインタラクティブに修正し、その結果を ML の学習データとして利用することで、ひび割れ自動検出精度を向上させる検出方法と、MRHMD によりひび割れ検出結果を強調して提示することで、人に見えにくい微細なひび割れを効果的に提示するシステムを提案する。

2 先行研究

2.1 ひび割れ画像の提示

本研究では、ひび割れ検出結果提示のために MRHMD を使用するが、一般的な MRHMD に搭載されている RGB-Depth (RGBD) カメラでは、検査に必要な微小なひび割れを検出ための解像度が不十分であるため、先行研究では外部の高解像度カメラを用いてひび割れ検出を行っている[8]。MRHMD では、RGBD カメラより習得した情報を用いて、検査対象の 3D データを作成し、システム起動時の MRHMD

の位置を原点とするワールド座標系を生成し、MRHMD の移動によって変化される RGBD 画像の変化と、加速度センサ情報を用いてローカル座標系を計算している。しかし、外部カメラで撮影した画像から作られたひび割れ画像を MRHMD 上で提示するためには、外部カメラ座標系をローカル座標系に変換するホモグラフィ変換行列の推定が必要である。ここで、ホモグラフィは三次元空間上にある物体を違う地点で見た場合に得られる画像間の座標関係を表す言葉である。先行研究では、ホモグラフィ変換行列推定のために特徴マッチングを行っている。

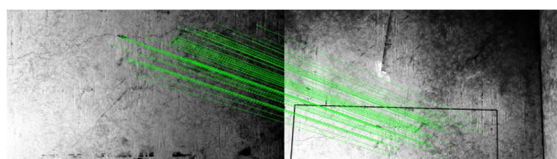


図 1: 特徴マッチングとホモグラフィ変換の例

特徴マッチングとは二つの画像間の類似度を比較する手法の一つである。図 1 のように特徴点抽出アルゴリズムより、比較したい画像から特徴点(角、パターンなど)を抽出し、各特徴点を Brute-Force Matcher (BF Matcher)より比較することで画像間のホモグラフィ変換行列を計算することが出来る[9]。ホモグラフィ変換行列は、Random Sample Consensus (RANSAC) アルゴリズムより計算される。ホモグラフィ変換は以下の式によって行われる。

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (1)$$

ここで、 x と y は入力画素の座標、 x' と y' は変換後の画素の座標、 h_{ij} はホモグラフィ変換行列の各要素を意味する。

上記の過程で得られるホモグラフィ変換行列は、外部カメラ座標系をローカル座標系に変換するものである。得られたホモグラフィ変換をひび割れ画像に適用することで、MRHMD 上で結果提示のためのひび割れ画像を作成することができる。ホモグラフィ変換されたひび割れ画像でプロジェクションマッピングを行うことで、ワールド座標系にひび割れ画像を投影することができる。先行研究では、プロジェクションマッピングの手法として UV 座標変換を用いている。しかし、撮影時の照明によって、ホモグラフィ変換行列推定が上手くいかない場合があるため、本研究では推定されたホモグラフィ変換が正しいかを評価する手法を提案する。

2.2 ひび割れ検出

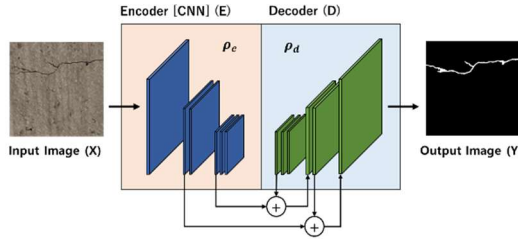


図 2: Encoder-Decoder based CNN の構造

Encoder-Decoder based CNN は入力画像と出力画像の画素数が同一であり, Image segmentation のために適したアルゴリズムとして用いられている. 図 2 で表しているように, Encoder-Decoder based CNN でエンコーダは, 過学習防止のため, 入力画像をダウンサンプリングし, 画像を分類する作業を行う. デコーダでは, ダウンサンプリングする時の情報を用いて, エンコーダで分類された出力を元の画素数に戻すアップサンプリングを行う過程である. Encoder-Decoder based CNN は, 以下の式でモデル化することが出来る.

$$Y = D(E(X; \rho_e); \rho_d) \quad (2)$$

ここで, X は入力の画像, E はエンコーダ, ρ_e はエンコーダパラメータ, D はデコーダ, ρ_d はデコーダパラメータ, Y は出力の画像である.

本研究では, Semantic segmentation の手法より入力画像をひび割れの画素とひび割れでない背景の画素に分類するために, Encoder-Decoder based CNN を使用する. ひび割れ検出のために利用する学習データは, 入力のコンクリート壁の画像と, それに対応するひび割れの位置をマークした Binary の Ground truth で構成される. しかし, Binary classification は誤検出を検知し難い問題がある.

また, ひび割れなど, 特定の画素分類用にモデルを最初から学習させるより, 一般的な物体認識用のイメージセットから学習されたモデルで転移学習を行うことが効率的であることが知られている[10]. したがって, 本研究では事前に学習されたモデルを用いてひび割れ検出用に転移学習を行う. また, 本研究では高解像度の画像を用いてひび割れ検出を行うが, 一般的な GPU ではメモリが不足しているため, 演算が不可能である. したがって, 入力画像を複数の小さい画像に分割し, 演算を行うことで GPU のメモリ不足の問題を解決する. 最終的には, 分割された複数のひび割れ検出結果画像を集め, 元の画像

の大きさに組み合わせることで, ひび割れの検出画像を生成する

3 提案手法

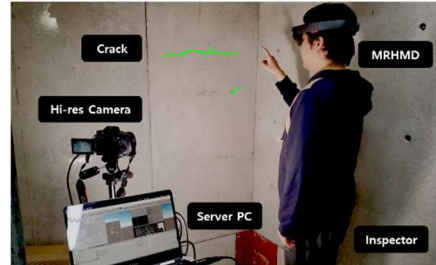


図 3: システム動作の様子

MRHMD を用いたひび割れ提示の先行研究[8]では, 事前に高解像度カメラで画像を撮影し, 別のシステムでひび割れを検出し, MRHMD のデータベースに画像を入力する必要がある, 検査時間が長くなる問題がある. したがって, 本研究では, 検査対象の撮影から結果の提示までに至る全体的なシステムを提案した. 図 3 のように, 現場に高解像度カメラと演算用のサーバーが設置された状況で, MRHMD を着用した検査員が検査対象の壁を指定することによって, リアルタイムで検査員に微小なひび割れを効果的に提示することができる.

また, Binary classification が誤検出に弱い問題を解決するために, 本研究では, 検査結果に誤検出がある領域を検査員が指定し, その情報を用いて検査環境に合わせたひび割れ検出用モデルを作成するインタラクティブなひび割れ検査システムを提案する.

3.1 システム構成

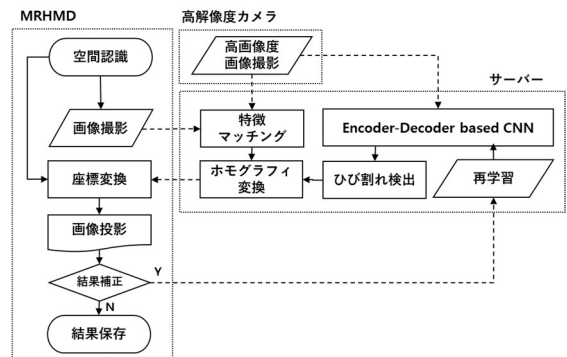


図 4: システムの流れ図

図 4 にシステムの処理の概要を示す. まず初めに MRHMD を用いて空間認識を行い, 検査現場の 3D

データを作成する。MRHMD より検索対象の画像撮影を行い、画像をサーバーに送信する。サーバー側では MRHMD から画像を受信した後、高解像度カメラで検査対象の高解像度画像撮影を行い、MRHMD で撮影された画像との特徴マッチングを行う。特徴マッチングよりひび割れ検出結果提示に必要なホモグラフィ変換行列を計算する。同時に、Encoder-Decoder based CNN で Semantic segmentation による高解像度の画像からひび割れ検出を行う。計算されたホモグラフィ変換をひび割れ画像に適用し、MRHMD に変換されたひび割れ画像を送信する。MRHMD では、検査現場の 3D データより得られる撮影時の位置と、検査対象間の位置関係より、画像投影の座標変換を行う。これらの過程で検索対象の画像からひび割れを検出し、MRHMD を用いて検査結果表示することができる。なお、検出されたひび割れ画像に誤検出が存在し、結果修正が必要な場合、検査員が誤検出のある領域を指定し、指定された領域は「誤検出」の画素として表示したデータで再学習を行うことで、ひび割れ検出精度を向上することができるシステムを実現する。

また、ひび割れ検出のために使われる Encoder-Decoder based CNN および検出結果提示のための特徴マッチングなどの演算量は膨大なため、モバイルプロセッサを使用している MRHMD だけでは演算に大きな時間がかかるため、サーバー上でこれらの演算を行うことによって、演算速度の高速化を行う。

3.2 ホモグラフィテスト

特徴マッチングより計算されるホモグラフィ変換は、2つの画像間で違う特徴点をマッチした場合、実際のホモグラフィ変換とは大きく異なるような変換行列を推定する場合がある。この問題を解決するために、本研究では、計算されるホモグラフィ変換が正しいかを判断するホモグラフィテストを提案する。

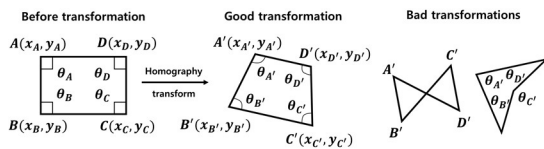


図 5: ホモグラフィ変換テストの例

ホモグラフィテストは、変換以前の画像の座標と変換後の画像の座標の位置関係を比較し、位置関係が保持される場合を成功と判断する。図 5 を参考に

して、ホモグラフィ変換以前の画像の位置関係は以下の式の通りである。

$$(x_A < x_D) \wedge (x_B < x_C) \wedge (y_B < y_A) \wedge (y_C < y_D) = P(3)$$

$$(\theta_A < \pi) \wedge (\theta_B < \pi) \wedge (\theta_C < \pi) \wedge (\theta_D < \pi) = R \quad (4)$$

$$P \wedge R = 1 \quad (5)$$

ここで、 A, B, C, D は入力画像の 4 つの頂点で、 θ_p は各頂点の内角を意味する。ホモグラフィ変換によって移動される各頂点に対しても式(5)が成り立つことを確認することで正しいホモグラフィ変換を判断することができる。

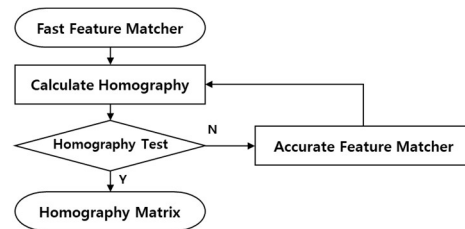


図 6: 順次的特徴点抽出アルゴリズム選択法

特徴マッチングで、画像間の特徴点が上手く検出できないと、ホモグラフィテストに通過できないため、本研究では、図 6 のように特徴点抽出時間が短いアルゴリズムで優先的にホモグラフィ変換行列を推定し、ホモグラフィテストに失敗した場合は、特徴点抽出精度の良いアルゴリズムでやり直す順次的特徴点抽出アルゴリズム選択法を提案する。

3.3 モデルの再学習と再初期化



図 7: 4-Class classification の概要

図 7 で示すように、True Positive (TP) と True Negative (TN) の 2 つに結果を分類する Binary classification は[6]、その構造は単純だが False Positive (FP) の情報が TP に、False Negative (FN) の情報が TN に分類される問題がある。本研究では、代案として FN と FP の情報を含めた 4-Class classification で Binary prediction を行う手法を提案する。

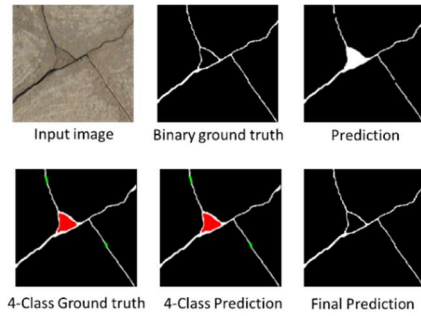


図 8: 4-Class Binary Prediction の概要

図 8 の Input image と Binary ground truth は、X. Yang[6]らによって作られたひび割れのデータセットである。Input image とひび割れ (TP) と背景 (TN) を白黒で表記する Binary ground truth より学習されたモデルは、Binary prediction データを生成することができる。しかし、Prediction にひび割れの誤検出 (FP) や未検出 (FN) が存在する場合がある。本研究では、Binary prediction 結果の TP と TN の情報に加えて、FP は緑、FN は赤として表記した 4-Class ground truth を用いて学習を行うことで学習結果補正を行う手法を提案する。4-Class ground truth で学習されたモデルは、4-Class prediction データを生成し、その結果の FN を TN のとして (黒)、FP を TP として (白) 表示することで Binary classification 問題を解決した Binary prediction を生成することができる。本研究では、上記の手法より、同一壁面に対するひび割れ検出精度が向上できるかどうかを実験で検証する。

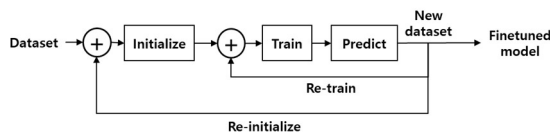


図 9: 4-Class Binary Prediction の概要

ここで、本研究では、上記の過程で生成された Ground truth でモデルを学習させる方法として再学習と再初期化を提案する。図 9 で示すように、初期化されたモデル上で新しい Ground truth を用いて学習する方法を再学習、初期化時のデータセットに加えて、最初からモデルを学習する方法を再初期化とする。

3.4 ひび割れ検出結果提示

検出されたひび割れ画像に対して、可視性の高い提示法として、ひび割れのある領域を緑色に変換して表示する方法を提案する。また、微小なひび割れ

は、高解像度カメラより撮影しても非常に小さく見えるため、効果的にひび割れ検出結果を提示する手法として、ひび割れに膨張の画像変換を適用してひび割れを大きく表示する手法を用いる。

4 システム開発

ここでは、提案システムの開発のために使用する機材やソフトウェア、MRHMD の詳細を記述する。サーバーと MRHMD 間の通信では TCP/IP プロトコルで Socket 通信を行い、大容量の画像を送受信する。

4.1 サーバー



図 10: サーバー用プログラムの起動画面

サーバーでは、特徴マッチングとホモグラフィ変換行列推定、Semantic segmentation とモデル再学習を行う。システムは Python 3.7 を用いて実装し、画像変換や特徴マッチングの演算には画像処理ライブラリーの OpenCV (4.1.2)を用いる。また、サーバーの GUI 実装のために PyQt5 を利用した。図 10 でこれらのサーバー用に作られたプログラムの起動画面を表す。

Semantic segmentation のために使用する Encoder-Decoder based CNN のアーキテクチャは、優秀な処理速度と精度を持っている LinkNet を使用する[11]。モデルのエンコーダとしては、学習効率が優れている EfficientNetB3 を用いる[12]。また、ImageNet より学習されたモデルを用いてひび割れ検出用に転移学習を行う[10]。モデルの Activation function は Softmax, Loss function は Dice loss を、Model weights は Adam optimizer により更新し、Learning rate は $1e-4$ から始め、Validation loss が 5 epochs の間で 0.05 以上変化がなかった場合、自動的に Learning rate を $1e-6$ まで減少させるアルゴリズムを使用した。これらのモデルは Keras 2.3 ライブラリーを用いて実装した。演算には、高性能 GPU

(GeForce RTX 2060 6GB)と CPU(Intel Core i7-6700, 3.4Ghz×8)とメモリ(DDR4 8GB×2)を用いた.

4.2 MRHMD

本研究では MRHMD 機器として Microsoft 社が開発した Hololens を使用する[13]. Hololens は空間認識用センサとしては 4 つのモノクロカメラ, 深度センサ, 加速度センサを, 画像処理用として RGB カメラを搭載している. また, ホログラフィックディスプレイを搭載しているため, あらゆる情報を 3D 空間上に投影して表示することができる. サーバーとの通信および本システムの実装には Unity 2018.4 と Visual Studio 2019 で Mixed Reality Toolkit 2.1 を用いて開発を行った.

4.3 モデル学習

一般的なひび割れ検出用にモデルを初期化するために X. Yang らによって提供されたひび割れのデータセット[6]を用いて学習を行った. データセットは 776 個のひび割れ画像と Ground truth のペアで, 256×256 px にリサイズし, 実際の学習に使われる Train データと, 学習の性能評価に使われる Valid データは, 7:3 の割合で学習を行った. また, ノイズに強いモデルを作るために Train データはアフィン変換や, ガウシアンフィルタなどを Data augmentation を適用して学習を行った[14]. 学習 Batch size は 2 で 50 世代(Epoch)学習を繰り返した. 毎世代学習の性能評価のためには Dice loss を, 学習されたモデルの評価のためには F1 score を測定した. Dice loss は以下の式で計算される.

$$DL(p, \hat{p}) = 1 - \frac{2 \sum p_{i,j} \hat{p}_{i,j}}{\sum p_{i,j} + \sum \hat{p}_{i,j}}, p \in \{0,1\}, 0 \leq \hat{p} \leq 1 \quad (6)$$

ここで p は Ground truth の値, \hat{p} は Prediction の値である, 上記の式より画像の i,j 成分に対する Dice loss を計算可能である. 1 世代の学習ごとに全体のデータセットに対する Dice loss が 0 に近いほど, よく学習できたと判断することができる. F1 score は以下の式で計算される.

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (7)$$

ここで TP は True positive, FP は False positive, FN は False negative の画素数を意味する. 1 世代の学習ごとに F1 score を計算し, 1 に近いほど, 学習されたモデルの精度がよいことを意味する.

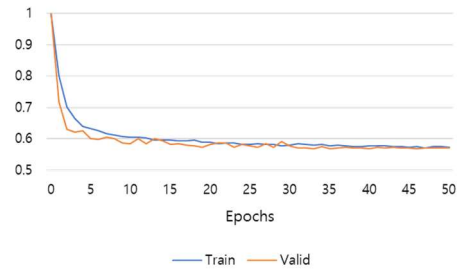


図 11: モデル学習世代による Dice loss の変化

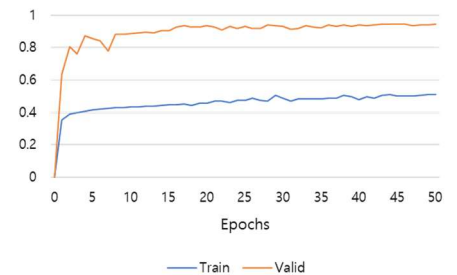


図 12: モデル学習世代による F1 score の変化

学習は合計で 50 世代行い, 各世代は 64.1 秒 (標準偏差 4.62 秒) がかり, 1 時間以内に学習は完了した. 学習結果は, 図 11 から分かるように, 学習を繰り返すほど, Dice Loss は減少し, 20 世代以降は大きい変化はなかった. 図 12 でも 20 世代以降には大きい変化がなかった, ここで, Train より Valid の F1 score が高い傾向をみせているが, これは Train データには学習を Robust にするために Data augmentation を適用しているが, 評価用の Valid では適用していないので, Valid の方の Prediction が上手くでき, F1 score が高く出ていると考えられる.

4.4 特徴点抽出アルゴリズムの比較

本システムで特徴マッチングを行うときに使用する特徴点抽出アルゴリズムを選択するために AKAZE[15], BRISK[16], ORB[17]の 3 つの特徴点抽出アルゴリズムを比較した. ここで, BRISK と ORB はパラメータ調整で特徴点検出数を変えることができるため, BRISK は Thresh を 20 (SLOW) と 40 (FAST) の 2 つ, ORB は Max features を 30000 (SLOW) と 10000 (FAST) の 2 つに分けて, 合計 5 個のアルゴリズムに対して比較を行った. 比較には研究室内部に設置されているコンクリート壁を 5m 距離で高解像度カメラより撮影した画像 (3840×2160 px, 200mm) と, それに対する地点を Hololens で 1m と 2m で撮影した画像を 3 パターン用意し, これらの組み合わせを 9 ヶ所撮影した画像を用意する. また, 高解像度カメラの画像と Hololens の画像

の Histogram を平滑化させることで、特徴点の検出率をよりよくすることができる[18].



図 13: Overlap error の可視化

上記のデータセットをアルゴリズムごとに 5 回繰り返し、ホモグラフィ変換行列を推定する性能を比較する。評価する項目は、全体データセットの中でホモグラフィテストに失敗したデータの数の Match fails, 計算時間の Computing time と Ground truth 推定ホモグラフィ変換のエラー率を表す Overlap error の 3 つの項目を評価する。Overlap error は以下の式で定義される[19].

$$E = 1 - \frac{N \cup N'}{N \cap N'} [\%] \quad (8)$$

ここで、 N と N' はそれぞれ Ground truth と推定されるホモグラフィ変換行列を適用した Binary mask の画素数である。図 13 のように、ホモグラフィ変換された Binary mask が重複している部分が少ないほど、Overlap error は大きくなる。

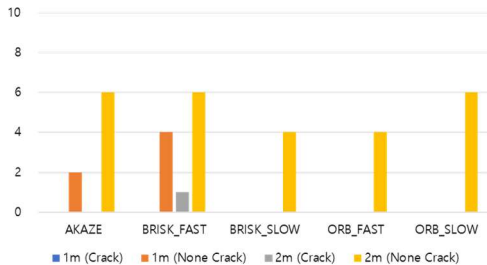


図 14: ホモグラフィ推定時の Match fails

図 14 から確認できるように Match fails に関しては、BRISK_FAST の場合、失敗率が一番多く、BRISK_SLOW と ORB_FAST が一番少なかった。しかし、どのアルゴリズムでも 2m 距離でひび割れのない壁面を撮影した画像に対しては失敗した。

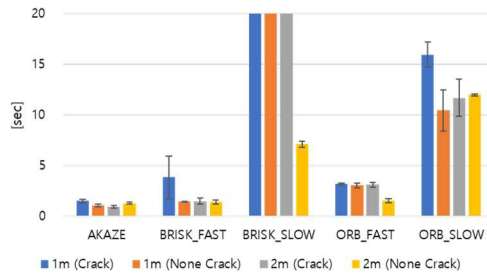


図 15: ホモグラフィ推定時の Computing time

図 15 の Computing time では、AKAZE が一番早く、BRISK_SLOW は 60 秒程度で一番遅かった。

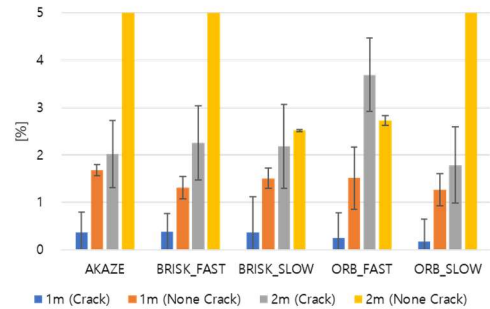


図 16: ホモグラフィ推定時の Overlap error

図 16 の Overlap error は、Match fails が起こった場合以外に関しては、ORB_SLOW が比較的に Overlap error が少なかった。上記の実験結果より、本研究では計算時間が少ない AKAZE を優先的に適応し、ホモグラフィテストを失敗した場合に限って、比較的に精度の良い ORB_SLOW で特徴マッチングをやり直す手法を用いる。

5 実験・結果

本研究では、作成したシステムの利用にあたって人が誤検出されたひび割れの領域を指定したデータでモデルを再学習させたときに、ひび割れの自動検出精度が向上するかどうかを評価する実験と、MRHMD によりひび割れ検出結果を強調して提示することによって、人に見えにくい微細なひび割れを効果的に提示することができるかの評価を実験で確認した。

5.1 ひび割れ検出精度の評価実験

本研究で提案した 4-Class Binary Prediction が、既存の Binary classification より良い学習成果を出せるかを比較する実験を行った。実験手順としては、最初に研究室内部に設置されているコンクリート壁を高解像度カメラで撮影し、3.2 でひび割れ検出用に初期化したモデルでひび割れ検出を行う。得られたひび割れ画像から誤検出は背景、未検出をひび割れとして表記することで新しい Binary ground truth を作成する。作られた Binary ground truth でモデルの F1 score を求め、比較の基準値とする。また、誤検出を赤、未検出を緑として記録した 4-Class ground truth を作成する。上記の過程より作られた Binary と 4-class の Ground truth を用いて再学習・再初期化を行う。再学習はひび割れ検出用に初期化

されたモデルを元に、新たに生成された Ground truth で転移学習をする過程とする。再初期化は、[6] のひび割れデータセットに、生成された Ground truth モデルを加えてモデル初期化過程である。これらの過程で作られたモデルで F1 score の比較を行う。

実験は、Training wall を対象に Ground truth を生成し、Training wall と同じコンクリート壁の違う場所を撮影した Inspection wall の F1 score で性能評価を行った。再学習および再初期化に使用した新しい画像は、 $3180 \times 2160\text{px}$ のイメージを $256 \times 256\text{px}$ に分割し、120 のデータセットを生成した。

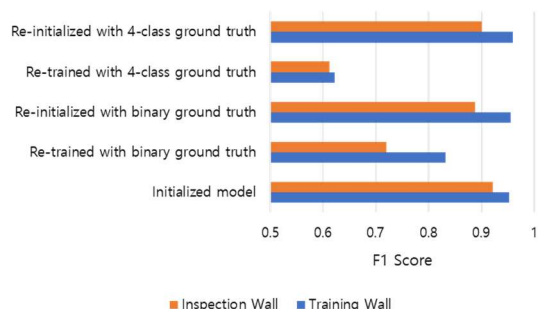


図 17: 再初期・再初期化時の F1 score の比較

図 17 から確認出来るように、再学習モデルは、Binary と 4-class どの場合でもひび割れ検出精度が低下した。再初期化モデルは、再学習モデルより検出精度が高いが、初期化モデルと大きい違いはなかった。図 18 は Training wall, 図 19 は Inspection wall に対して、各モデルでひび割れ検出を行った結果の画像である。初期化モデルと再初期化モデルの比較より分かるように、再初期化モデルでは、ひび割れではない成分が上手く除去されることがわかる。Inspection wall では、初期化モデルの場合、コンクリート壁の角をひび割れとして誤検出しているが、再学習モデルでは壁の角を誤検出として検出していることがわかる。

5.2 ひび割れ検出結果提示の評価実験

本システムでは、高解像度カメラより検出されたひび割れ画像を、MRHMD を用いて提示することをを行った。MRHMD で結果提示を行う場合、4-class の結果の赤は黒、緑は白ピクセルに変換して表示し、ひび割れを緑色に色付けし、膨張画像変換を適用して表示する提示法の性能評価を行うために、本システムを使用した場合と、使用しない場合のコンクリート壁の画像を Hololens より撮影し、比較を行った。

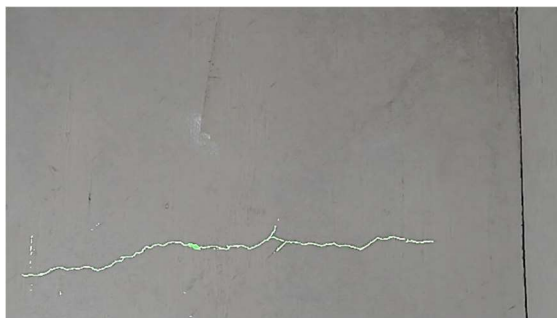


図 20: MRHMD でのひび割れ検出結果の提示



図 21: MRHMD での壁面撮影画像

図 20 は Hololens 上で本システムのひび割れ可視化（緑色に表示、膨張変換）手法を適用した場合の画像で、図 21 は適用しなかった場合の Hololens 撮影画像である。2つの図よりわかるように、ひび割れ可視化手法を用いた場合、明らかにひび割れを確認することができる。

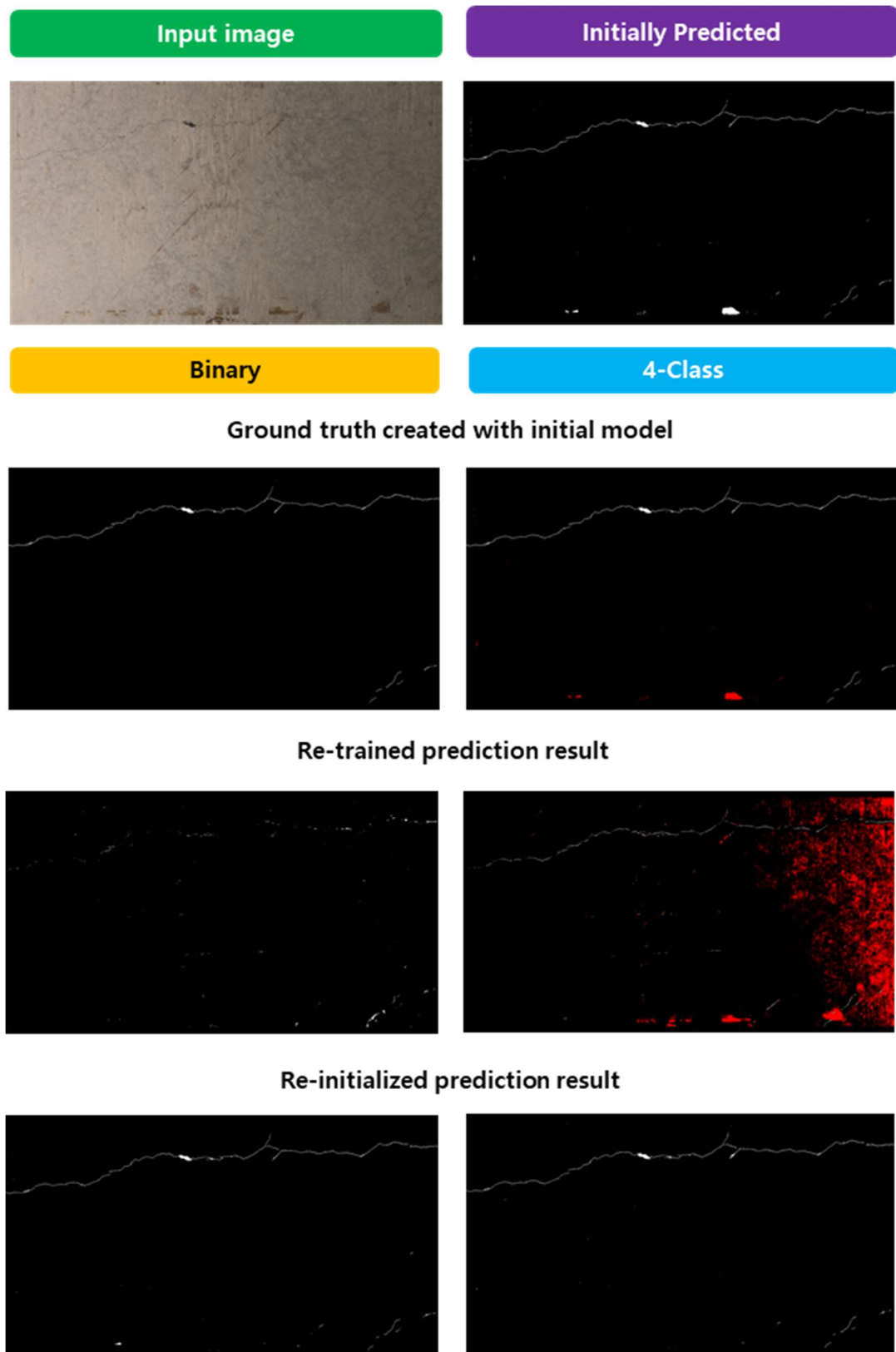


図 18: 初期化されたモデルでひび割れを検出し, 新しい Binary と 4-Class の Ground truth を作って再学習・再初期化させたときのひび割れ検出結果

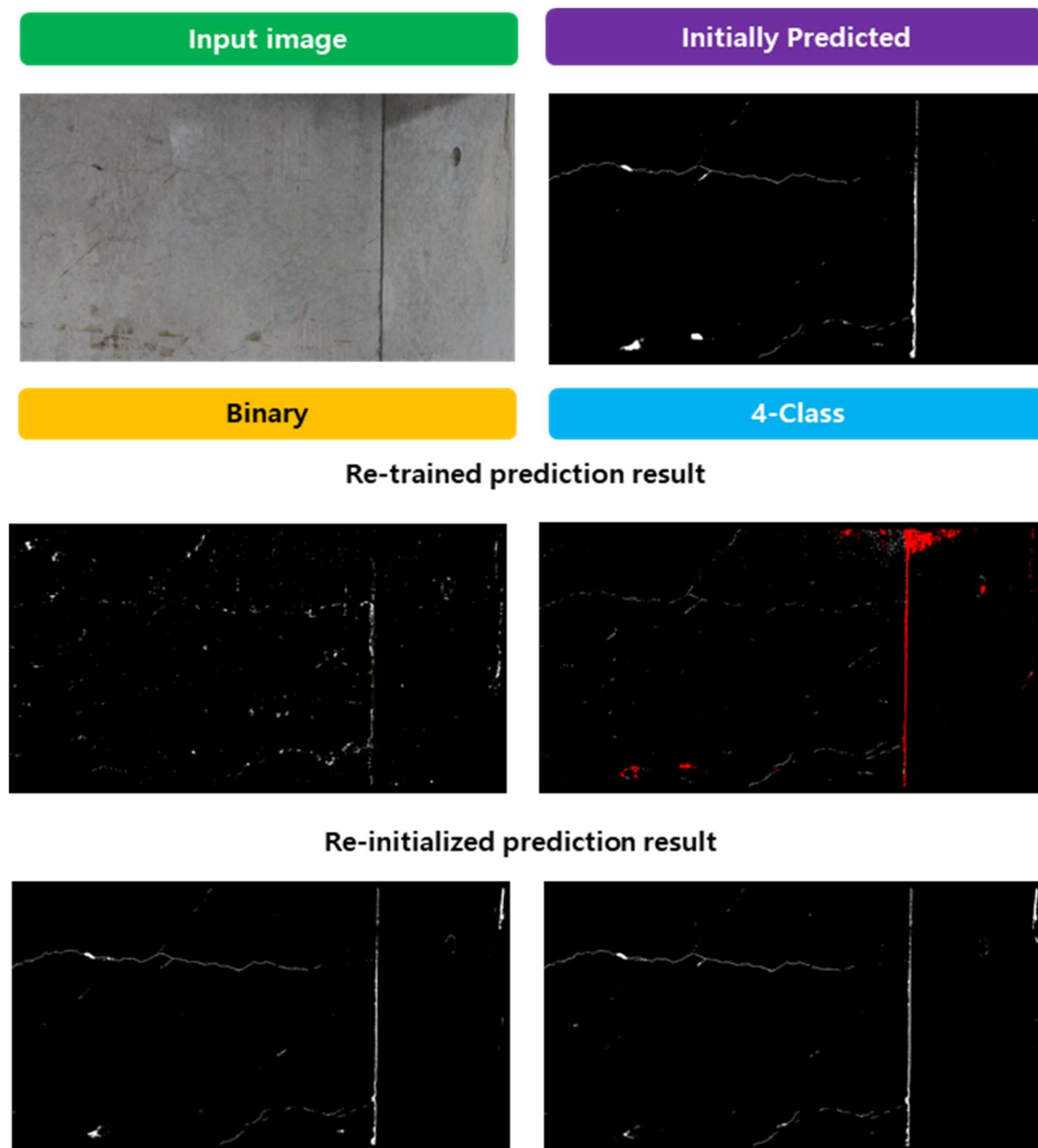


図 19: 同じ壁面で違う場所の Ground truth で再学習・再初期化させたときのひび割れ検出結果

6 考察

5.1 のひび割れ検出精度の評価実験の結果である図 17 より, 再学習が逆にモデルの精度を低下させることを確認した. これより, 最初のモデル初期化作業がモデルの精度向上に大きく影響を与えと考えられる. しかし, 新しい Ground truth を生成し, モデルを再初期化した場合, 図 20 より, 初期化モデルでは誤検出されたノイズがよく除去できることから, 検査対象のコンクリート壁面に対して再初期化

を行う手法は有効であると考えられる. 再学習で上手く行かなかった理由は, 少ないデータセットで学習をさせ, モデルが過学習されたと考えられる. 再学習に短い時間がかかり, 精度の向上が期待できないことに比べて, 再初期化は 1 時間前後の学習時間がかかり, ひび割れ検出精度を向上できる方法として考えられる.

5.2 のひび割れ検出結果提示の実験の結果である図 18 では, ひび割れに画像変換を適用して, MRHMD より提示することで, 人が見にくい微小な

ひび割れを効果的に提示することができることを確認できる。

7 まとめ

本研究では、Semantic segmentation 手法によって検出されたひび割れ画像に対して、人が誤検出の領域を指定することで作られたデータをモデルの学習データとして利用し、ひび割れ検出精度を向上させる手法と、MRHMD を用いてひび割れ画像を強調して表示することで、微小なひび割れを効果的に提示するシステムを提案し、そのシステムの性能評価を行った。高解像度カメラより撮影した画像からひび割れを検出し、MRHMD より撮影された画像と特徴マッチングを行うことで、MRHMD 上にひび割れ画像を投影して提示することができた。

特徴点検出アルゴリズムとしては比較的早く、精度の良い AKAZE を優先的に適用し、ホモグラフィテストでホモグラフィ変換行列推定が正しく行われたかを確認する。失敗した場合は、ORB_SLOW よりホモグラフィ行列推定をやり直すことで、早くてロバストにホモグラフィ変換を推定することができた。

ひび割れ画像検出アルゴリズムは、Encoder-Decoder based CNN を用いた Semantic segmentation の手法より、1 時間弱の学習時間で精度の良い (F1 score > 0.95) ひび割れ検出用の初期化モデルを生成した。多様な検査現場の環境に応じて、検査員が誤検出と未検出の領域をインタラクティブに指定することで誤検出に強いひび割れ検出モデルを作るために 4-Class Binary Prediction による再学習・再初期化の手法を提案した。再学習の場合は、Binary と 4-class の Ground truth どちらも性能の低下が行われた。しかし、再初期化では、初期化モデルより誤検出をよく除去できるようになった。

ひび割れ検出結果提示には、ひび割れ画像に色付けや膨張変換を行うことで、微小なひび割れを人間に効果的に提示することができることを確認した。

現世代の MRHMD デバイスの性能が不足しているため、本研究ではサーバーとの通信でひび割れ検出を行ったが、技術が発展し、MRHMD のモバイルプロセッサとカメラの性能向上が実現された場合、本研究で提案された手法は有効であると考えられる。検索対象に接近せずに、ひび割れを検出し、確認できる本システムは、コンクリート橋など、接近しにくい環境で役立つと考えられる。

8 謝辞

卒業研究、卒業論文執筆にあたって、ご家族の皆様と多くの指導をしてくださった山口先生と、卒業論文の推敲に多くご協力をしていただいた西村先輩、檜原先輩、危さんと山本君に感謝の意を表する。

参考文献

- [1] 国土交通省, “建設業を取り巻く情勢・変化参考資料,” [Online]. Available: <https://www.mlit.go.jp/common/001121700.pdf>. [Accessed: 05-Feb-2020].
- [2] 道路構造物管理研究室, “PRC 道路橋の性能照査に関する研究,” 国土交通省. [Online]. Available: <http://www.nilim.go.jp/lab/bcg/siryou/tnn/tnn0620.htm>. [Accessed: 05-Feb-2020].
- [3] 国土交通省, “道路トンネル点検要領,” [Online]. Available: https://www.mlit.go.jp/road/sisaku/yobohozen/tenken/yobo3_1_9.pdf. [Accessed: 05-Feb-2020].
- [4] A. Mohan and S. Poobal, “Crack detection using image processing: A critical review and analysis,” Alexandria Engineering Journal, vol. 57, no. 2, pp. 787–798, Jun. 2018.
- [5] Y. J. Cha, W. Choi, and O. Büyüköztürk, “Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks,” Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 5, pp. 361–378, 2017.
- [6] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, “Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network,” Computer-Aided Civil and Infrastructure Engineering, vol. 33, no. 12, pp. 1090–1109, Aug. 2018.
- [7] H.-G. Moon and J.-H. Kim, “Intelligent Crack Detecting Algorithm on the Concrete Crack Image Using Neural Network,” 28th International Symposium on Automation and Robotics in Construction (ISARC 2011), 2011.

- [8] T. Yamaguchi, T. Shibuya, M. Kanda, and A. Yasojima, "Crack Inspection Support System for Concrete Structures Using Head Mounted Display in Mixed Reality Space," SICE Annual Conference 2019 (SICE 2019), Hiroshima, Japan, Sep. 2019.
- [9] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Mar. 2018.
- [10] S. Kornblith, J. Shlens, and Q. V. Le, "Do Better ImageNet Models Transfer Better?," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
- [11] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," 2017 IEEE Visual Communications and Image Processing (VCIP), 2017.
- [12] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," arXiv preprint arXiv:1905.11946, 2019.
- [13] Microsoft, "Microsoft HoloLens | Mixed Reality Technology for Business," [Online]. Available: <https://www.microsoft.com/en-us/hololens/>. [Accessed: 05-Feb-2020].
- [14] T. S. Kumar, "Enhancement of Image Classification through Data Augmentation using Machine Learning," International Journal of Computer Sciences and Engineering, vol. 6, no. 9, pp. 220–224, 2018.
- [15] P. F. Alcantarilla et al., "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in British Machine Vision Conference, Bristol, BMVC, 2013.
- [16] S. Leutenegger et al., "BRISK: Binary robust invariant scalable keypoints," in IEEE International Conference on Computer Vision, Barcelona, ICCV, 2011, pp. 2548–2555.
- [17] E. Rublee et al., "ORB: An efficient alternative to SIFT or SURF," in IEEE International Conference on Computer Vision, Barcelona, ICCV, 2011, pp. 2564–2571.
- [18] M. M. Daud, Z. Kadim, S. L. Yuen, H. H. Woon, I. Faye, and A. S. Malik, "A pre-processing approach for efficient feature matching process in extreme illumination scenario," 2012 3rd International Conference on Image Processing Theory, Tools and Applications (IPTA), 2012.
- [19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool, "A Comparison of Affine Region Detectors," International Journal of Computer Vision, vol. 65, no. 1–2, pp. 43–72, 2005.