EfficientPose: An efficient, accurate and scalable end-to-end 6D multi object pose estimation approach

Abst

本論文では，6次元物体の姿勢を推定するための新しい手法であるEfficientPoseを紹介する．この手法は，高い精度，効率性，そして幅広い計算機資源に対するスケーラビリティを備えています．さらに，EfficientPoseは，複数のオブジェクトやインスタンスの2Dバウンディングボックスを検出するだけでなく，それらの6Dポーズをワンショットで推定することができる．これにより，他のアプローチが抱える，複数のオブジェクトを扱う際のランタイムの大幅な増加を解消することができます． これらのアプローチは，まずキーポイントなどの2Dターゲットを検出し，その後，各オブジェクトの6Dポーズについ てPerspective-n-Point問題を解くことを目的としています． 我々のアプローチは，広く使われている6Dポーズ推定のベンチマークデータLinemodのRGB入力に対して，ADD(-S)指標で97.35%という最先端の精度を達成し，しかも27 FPS以上でエンドツーエンドで動作する．また，複数のオブジェクトやインスタンスを扱うことができ，シングルショットの2Dオブジェクト検出と6Dポーズ推定を融合させることで，複数のオブジェクト（8個）があっても26 FPS以上でエンド・ツー・エンドで動作することができ，現実世界の多くのシナリオにとって非常に魅力的な手法となっています． コードは以下のサイトで公開されます．

Intro

画像中の興味ある物体を検出することは，コンピュータビジョンの重要なタスクであり，この研究分野では，この問題に取り組むための高精度な手法が多く開発されている[27][8][45][21][32]．最近では，計算機や実行時間に制限のある実世界のシナリオに適用できるように，精度だけでなく効率性にも着目した研究もあります[41][38]．例えば，Tanら[38]は，単一のハイパーパラメータで，高い範囲の計算資源，速度，精度に容易に対応できるEfficientDetと呼ばれる非常にスケーラブルで効率的なアプローチを開発しました．しかし，ロボット操作，自律走行車，AR（拡張現実）などのタスクでは，画像内のオブジェクトの2Dバウンディングボックスを検出するだけでは不十分で，6Dポーズも推定する必要があります． RGB入力の6次元物体姿勢推定の分野で最先端の精度を達成している最近の研究の多くは，画像内の注目物体の2次元ターゲット（キーポイントなど）を最初に検出し，その後にPnPアルゴリズムを用いて6次元姿勢を解くアプローチに依存している[39][25][26][44][20][35].

これらの手法は，6 次元ポーズ推定の精度が高く，また，単一物体のポーズ推定に関しては非常に高速なものもあるが，物体の数が増えると実行時間が線形に増加する．これは，各オブジェクトに対して個別にPnPによる6次元ポーズを計算する必要があることに起因します．さらに，いくつかの手法では，必要なキーポイントを検出するために，ピクセル単位のRANSACベース[10]の投票スキームを使用していますが，これも各オブジェクトに対して個別に実行する必要があり，非常に時間がかかる可能性があります[26][35]． さらに，いくつかの手法では，関心のあるオブジェクトのバウンディングボックスをローカライズおよびクロッピングするために，最初に別の2Dオブジェクト検出器を必要とします．これらのクロッピングされた画像パッチは，その後，実際の6Dポーズ推定アプローチの入力として使用されるため，検出されたオブジェクトごとに手法全体を個別に適用する必要がある[25][20]． これらの理由により，これらのアプローチは，複数のオブジェクトを使用するユースケースには適していないことが多く，ランタイムの制限により，多くの実世界のシナリオでの展開が阻害されています．

本研究では，これらの問題に対処することなく，広く使用されているベンチマークデータセット「Linemod」[15]のRGB入力を用いて，最先端の性能を達成する新しいアプローチを提案する．これを実現するために，我々は最先端の2Dオブジェクト検出アーキテクチャであるEfficient-Detsファミリを直感的な方法で拡張し，オブジェクトの6Dポーズも予測する． そのため，オブジェクトの移動と回転を予測するために，クラス分類とバウンディングボックス回帰のサブネットに類似した2つのサブネットを追加します．これらのサブネットは比較的小さく，入力フィーチャーマップの計算を既存のネットワークと共有しているため，追加の計算コストをかけずに，非常に簡単に6次元のフルポーズを取得することができます． EfficientDetアーキテクチャーとのシームレスな統合により，我々のアプローチは，複数のオブジェクトカテゴリや複数のオブジェクトインスタンスを検出し，それらの6Dポーズを推定することも可能です．また，6次元ポーズを直接推定するため，RANSACやPnPなどの後処理も必要ありません． これにより，本手法の実行時間は，画像中のオブジェクト数にほとんど依存しない．

LinemodデータセットにおいてADD(-S)指標で最先端の精度が報告されたのは，我々が提案した6Dオーグメンテーションが重要な要素であることが判明しました． この提案されたオーグメンテーション技術により，我々のような直接的な6Dポーズ推定法は，画像の回転や拡大縮小も使用できるようになります．これは，そうでなければ，画像と注釈されたポーズの間にミスマッチが生じてしまいます．このような画像操作は，Linemod[6][45]のような小さなデータセットを扱う際に，性能と一般化を大幅に改善するのに役立ちます． 2D+PnPアプローチは，2Dターゲットが画像変換に応じて比較的容易にトランスフォームされるため，これらの方法を無理なく利用することができる．我々の提案する拡張手法を用いることで，2D+PnPアプローチのこれまでの優位性を補うことができる．これが，現在，RGB入力による6Dオブジェクトポーズ推定の分野で2D+PnPアプローチが優勢である理由であると考えられる[26][44][35].

オリジナルのEfficientDetsと同様に，我々のアプローチもまた，単一のハイパーパラメータ ϕ によって高い拡張性を持ち，ネットワークを幅広い計算資源，速度，精度に合わせて調整することができま

す．また，本手法は，前述のように後処理を必要とせず，複数のオブジェクトカテゴリーやインスタンスを扱うことができるアーキテクチャに基づいているため，非常に使いやすく，多くの実世界のシナリオにとって魅力的な手法となっています．

結論から言うと，この作品における私たちの主な貢献は以下の通りです．

- 直接的な6次元ポーズ推定のための6Dオーグメンテーションは，特に小さなデータセットを扱う場合に，性能と汎用性を向上させるためのアプローチです．
- EfficientDetsの最先端の2Dオブジェクト検出ファミリーを拡張し，6Dオブジェクトのポーズ推定機能を追加しました．シングルショットのマルチオブジェクトやインスタンスの検出，高精度，スケーラビリティ，効率性，使いやすさといったEfficientDetsの利点はそのままです．

3.Methods

ここでは，RGB画像を入力として，6次元物体のポーズを推定する手法について説明する． 6Dポーズは，物体の3次元回転 $R \in SO(3)$ と3次元平行移動 $t \in \mathbb{R}^3$ の2つの部分から構成され，物体座標系からカメラ座標系への剛体変換を表します．このタスクには，2D画像内の物体を最初に検出することや，複数の物体カテゴリーやインスタンスを扱うことなど，いくつかのサブタスクが含まれており，これらは比較的成熟した2D物体検出の分野の最近の研究ですでに解決されているため，私たちはこのような2D物体検出アプローチをベースに，物体の6Dポーズを予測する機能を追加することにしました．

3.1. Extending the EfficientDet architecture

我々の目標は，EfficientDetのアーキテクチャを直感的な方法で拡張し，計算オーバーヘッドを小さく保つことです．そのため，分類と境界ボックス回帰のサブネットワークに類似した2つの新しいサブネットワークを追加しますが，各アンカーボックスのクラスと境界ボックスオフセットを予測する代わりに，新しいサブネットはそれぞれローテーション R とトランスレーション t を予測します． これらのサブネットは規模が小さく，既存の分類サブネットやボックスサブネットと入力フィーチャーマップを共有しているため，計算コストの増加は最小限に抑えられます． この2つのサブネットを介して6次元ポーズ推定のタスクを統合し，ベースアーキテクチャのアンカーボックスマッピングとNMS（Non-Maximum Suppression）を使用して，背景や複数のデセクションをフィルタリングすることで，6次元ポーズを検出できるアーキテクチャを作成することができました．

- Class
- 2D bounding box
- Rotation

- Translation

の1つまたは複数のオブジェクト・インスタンスとカテゴリーを1回の撮影で取得します． 基盤となるEfficientDetアーキテクチャのスケーラビリティを維持するために，ローテーションおよびトランスレーション・ネットワークのサイズもスケーリング・ハイパーパラメータ ϕ によって制御されます．図2は，EfficientDetのアーキテクチャの概要を示したものです．基本アーキテクチャの詳細については，EfficientDetの出版物[38]を参照してください．

3.2. Rotation Network

回転に軸角表現を選んだのは，四元数よりもパラメータが少なくて済むことと，Mahendranら[24]の実験でも若干良い結果が得られたからです．また，Mahendranら[24]の実験では，わずかに良い結果が得られています．しかし，この表現は我々のアプローチにとって重要ではなく，必要に応じて切り替えることができます．したがって，サブネットワークは，回転行列$R \in SO(3)$ の代わりに，各アンカーボックスに対して1つの回転ベクトル $r \in \mathbb{R}^3$ を予測します．ネットワークのアーキテクチャは，EfficientDet[38]の分類およびボックスネットワークに似ていますが，出力された r_{init} を直接回帰された回転として使用する代わりに，Kanazawaら[17]にインスパイアされた反復精製モジュールを追加します． このモジュールは，現在の回転 r_{init} のチャネル次元に沿った連結と， r_{init} を入力として出力する最初の回帰層の前の最後の畳み込み層の出力を取り，Δr を回帰して，最終的な回転回帰が

$$r = r_{init} + \Delta r \tag{1}$$

反復精製モジュールは，深さ方向に分離可能な畳み込み層 D_{iter} [5]と，それに続くグループ正規化[42]およびSiLU（swish-1）活性化関数[29][9][14]から構成されています． 層の数 D_{iter} は，スケーリング・ハイパーパラメータ ϕ に依存し，以下の式で表されます．

$$D_{iter}(\phi) = 2 + \lfloor \phi/3 \rfloor \tag{2}$$

ここで，$\lfloor \rfloor$ はフロア関数を表す．これらの層の後には，出力層（線形活性化関数を持つ単一の深さ方向に分離可能な畳み込み層）が続き，Δr を出力します．

この反復精製モジュールは，回転 r にN_{iter} 回適用され，ベースネットワークの出力 r_{init} で初期化され，各中間反復ステップの後，r は次のステップのためにr_{init} に設定されます．N_{iter} は，スケーラビリティを維持するために ϕ にも依存し，次のように定義されます．

$$N_{iter}(\phi) = 1 + \lfloor \phi/3 \rfloor \tag{3}$$

すべての層のチャンネル数は，出力層を除き，クラスネットワークやボックスネットワークと同じで，アンカーの数と回転パラメータによって決定される．式2および式3は，EfficientDet[38]に掲載さ

れているボックスネットワークとクラスネットワークの深さ D_{box} および D_{class} の式に基づいていますが，さらなる実験による裏付けはなく，最適化される可能性もあります． 完全な回転ネットワークのアーキテクチャを図3に，絞り込みモジュールの詳細なトポロジーを図4に示します．

図3 初期回帰モジュールと反復改良モジュールを備えた回転ネットワークアーキテクチャ．各畳み込みブロックは，深さ方向に分離可能な畳み込み層と，それに続くグループ正規化およびSiLU活性化で構成されています．

サブセクション3.3で説明した回転・並進ネットワークの設計は，バニラのEfficientDetのボックス・クラスネットワークに基づいていますが，学習時に必要な最小バッチサイズを減らすために，バッチ正規化をグループ正規化に置き換えています[42]．この置き換えにより，バッチサイズ1で回転および平行移動ネットワークをスクラッチからトレーニングすることに成功し，バッチ正規化で必要な最小バッチサイズ32と比較して，トレーニング時に必要なメモリ量を大幅に削減することができました．Wuら[42]によると，1グループあたり16チャネルを目標にしているので，グループ数 N_{groups} は以下のように計算されます．

$$N_{groups}(\phi) = \lfloor \frac{W_{bifpn}(\phi)}{16} \rfloor \tag{4}$$

ここで，W_{bifpn} はEfficientDetのBiFPNと予測ネットワークのチャンネル数を表しています[38].

3.3. Translation Network

並進ネットワークのネットワークトポロジーは，3.2節で説明した回転ネットワークと基本的に同じであり，各アンカーボックスに対して並進 $t \in R3$ を出力する点が異なる．しかし，並進ベクトル $t = (t_x, t_y, t_z)^T$ の全成分を直接回帰するのではなく，PoseCNN[43]のアプローチを採用し，物体の2次元中心点 $c = (c_x, c_y)^T$ をピクセル座標で予測するタスクと，距離 t_z を個別に予測するタスクに分けている．中心点 c，距離 t_z，カメラ固有のパラメータがあれば，並進 t の欠落成分 t_x，t_yは，ピンホールカメラを想定して，以下の式で計算できる．

$$t_x = \frac{(c_x - p_x) \cdot t_z}{f_x} \tag{5}$$

$$t_y = \frac{(c_y - p_y) \cdot t_z}{f_y} \tag{6}$$

ここで，$p = (p_x, p_y)^T$ は主点，f_x, f_yは焦点距離です．

各アンカーボックスに対して，そのアンカーボックスの中心から，対応するオブジェクトの中心点までのオフセットをピクセル単位で予測します．これは，図5に示すように，現在の点から中心点までのオフセットを予測することと同じである．相対的な空間関係を維持するために，このオフセットは，特徴ピラミッドの各レベルからの入力特徴マップのストライドで正規化される．これを用いて

- 予測される相対的なオフセット
- 各点がそれぞれのx，y座標を持つ，特徴量マップの座標マップX，Y
- そして，ストライド

の絶対座標を算出することができる． ここでの意図は，絶対座標 c_x，c_y を直接回帰するのではなく，畳み込みの並進不変性を利用して，特徴量マップの各点における相対的なオフセットをネットワークが予測する方が簡単ではないかということである．また，この仮定を実験的に検証した．

上述した 2D 中心点からの移動量 t および奥行き tz の計算，ならびに予測された相対的なオフセットからの絶対的な中心点座標 cx および cy の計算は，余分な後処理を避け，GPU または TPU による加速を可能にするために，アーキテクチャを可能な限りシンプルに保ちながら，いずれも別の TensorFlow[1] 層に実装されています．先に述べたように，t の計算には固有のカメラパラメータも必要であり，これが翻訳ネットワークに必要な別の入力層がある理由です．この入力層は，ピンホールカメラの焦点距離 f_x と f_y，主点座標 p_x と p_y，そして最後にオプションの平行移動係数 $s_{translation}$ と画像スケール s_{image} を含むベクトル $a \in \mathbb{R}^6$ を各入力画像に提供します．並進スケーリング係数 $s_{translation}$ は，例えばmmからmへの並進単位を調整するために使用することができます．画像スケール s_{image} は，t を回復するために式5および式6を適用するために，予測された中心点 c を元の画像解像度に再スケールするのに必要な，元の画像サイズから入力画像サイズへのスケーリング係数です．

3.4. Transformation Loss

使用する損失関数は，PoseCNN[43]の PoseLoss と ShapeMatch-Loss をベースにしていますが，回転のみを考慮するのではなく，本アプローチでは並進も考慮しています．非対称な物体に対して，我々の損失 L_{asym} は以下のように定義される．

$$L_{asym} = \frac{1}{m} \sum_{x \in M} ||(Rot(\tilde{r}, x) + \tilde{t}) - (Rot(r, x) + t)||_2 \tag{7}$$

ここで，$Rot(r, x)$ と $Rot(\tilde{r}, x)$ はそれぞれ，x をグランドトゥルースの回転 r とRodriguesの回転式[7][34]を適用した推定回転 $\tilde{r}$ で回転させることを示す． さらに，M は物体の3次元モデルの点の集合を示し，m は点の数である．損失関数は，基本的に，対象物のグランドトゥルース6次元ポーズと推定6次元ポーズの変換を行い，変換されたモデル点間の平均点距離を計算するもので，4.2項で説明したADDメトリックと同じものです．この方法では，パフォーマンスを測定する指標に基づいてモ

デルが直接最適化されるという利点があります． また，回転と平行移動の損失が互いに独立して計算される場合，部分的な損失のバランスをとるために余分なハイパーパラメータを使用する必要がありません．

対称的な物体も扱うため，対応する損失 L_{sym} は以下の式で与えられます．

$$L_{asym} = \frac{1}{m} \sum_{x_1 \in M} \min_{x_2 \in M} ||(Rot(\tilde{r}, x_1) + \tilde{t}) - (Rot(r, x_2) + t)||_2 \tag{8}$$

これは L_{asym} と似ていますが，2つの変換点セットのマッチング点間の距離を厳密に計算する代わりに，各点が他の変換点セットの任意の点との間の最小距離を考慮しています．これにより，Xiangら[43]が述べているように，対称的なオブジェクトを扱う場合に，トレーニング中の不必要なペナルティを避けることができます．

完全変換損失関数 L_{trans} は以下のように定義されます．

$$L_{trans} = \begin{cases} L_{sym} \; if \; symmetric, \\ L_{asym} \; if \; asymmetric. \end{cases} \tag{9}$$

3.5. 6D Augmentation

今回使用したLinemod[15]およびOcclusion[2]データセットは，アノテーションデータの量が非常に限られています．Linemodは，1つのオブジェクトに対して約1200のアノテーションが施された例で構成されており，Occlusionは，1つのシーンのすべてのオブジェクトがアノテーションされたLinemodのサブセットであるため，データ量は同様に少なくなります．このため，大規模なニューラルネットワークでは，より一般的な解に収束させることが特に難しくなります．このようなシナリオでは，データの増強が大いに役立ち[6][45]，あらゆる2D検出とPnPアプローチに依存する手法がここでは大きな利点となります．このような手法は，キーポイントなどの2Dターゲットが画像変換に応じて比較的容易に変換されるため，回転，拡大縮小，剪断などの画像操作技術を容易に使用することができます．しかし，物体の6次元姿勢を直接予測する手法は，回転などの画像変換によって，画像とグランドトゥルースの6次元姿勢との間にミスマッチが生じるため，この面では限界がある．この問題を解決するために，私たちは，画像をランダムに回転および拡大縮小し，グランドトゥルースの6次元ポーズを拡張画像と一致するように変換することができる6次元拡張を開発しました．図6に示すように，主点を中心に画像を角度 $\theta \in [0^\circ, 360^\circ)$ で2次元回転させると，6次元ポーズの3次元回転Rと並進 t もz軸を中心に θ 回転させる必要がある．このz軸周りの回転は，軸角表現の回転ベクトル Δr で以下のように記述できる．

$$\Delta r = \left(0, 0, \frac{\theta}{180 \cdot \pi}\right)^T$$

Δr から得られた回転行列 ΔR を用いて，補強された回転行列 R_{aug} と並進 t_{aug} は以下の式で計算されます．

$$R_{aug} = \Delta R \cdot R$$

$$t_{aug} = \Delta R \cdot t$$

画像のスケーリングを補強技術としても扱うためには，並進 $t = (t_x, t_y, t_z)^T$ の t_z 成分を調整する必要があります．画像をファクタ f_{scale} で再スケーリングすると，拡張された並進 t_{aug} は次のように計算されます．

$$t_{aug} = \left(t_x, t_y, \frac{t_z}{f_{scale}} \right)$$

注意しなければならないのは，対象物が画像の中心にない場合，スケーリングによる拡張はエラーになるということです．画像をリスケールしても，対象物の2D投影は変わらず，大きくなったり小さくなったりするだけです．大きくなったり小さくなったりするだけです．しかし，実際にはZ軸に沿ってオブジェクトを移動させると，カメラから3Dオブジェクトへの視界が変わり，2D画像平面への投影も変わってしまいます． しかし，4.7節に示すように，この拡張技術によって得られる追加データの利点は，導入された誤差を強く上回っています．図7は，左上の画像がグランドトゥルースの6Dポーズを持つオリジナルの画像で，その他の画像はこのサブセクションで説明する方法で補強されている例です．この作品では，すべての実験に，区間 $\theta \in [0^\circ, 360^\circ)$ から一様にサンプリングされたランダムな角度 θ と，$[0.7, 1.3]$ から一様にサンプリングされたランダムなスケーリングファクター f_{scale} を使用しています．

3.6. Color space Augmentation

また，注釈付きの6次元ポーズを調整することなく適用できる，色空間におけるいくつかの補強技術を使用している．このタスクでは，R and Augment[6]法を採用しています．これは，複数のデータセットやモデル間で性能を向上させ，汎用性を高めることができる学習済みの補強法です．R and Augment法は，入力画像のコントラストや明るさを調整するなど，複数の拡張手法から構成されており，適用する画像変換の数nと変換の強さmの2つのパラメータで調整することができます．

前述したように，回転や剪断などの一部の画像変換は，入力画像とグランドトゥルースの6Dポーズとの間にミスマッチが生じるため，R and Augment法からそれらのオーグメンテーション技術を削除します． さらに，ガウシアンノイズを加えて選択します．増幅強度をパラメータ m で設定するアプローチを維持するために，チャンネル単位の加法性ガウスノイズを，範囲 $[0, \frac{m}{100} \cdot 255]$ の正規分布か

らサンプリングします．実験では，各画像について，整数の一様分布 $[1, 3]$ から無作為に抽出した n と，$[1, 14]$ から抽出した m を用いた．

4. Experiments

このセクションでは，我々が行った実験，実験のセットアップと実装の詳細，および使用した評価指標について説明します． また，Linemod実験では，我々の結果を現在の最先端の手法と比較しています．なお，我々の手法は，$\phi = 0$ から $\phi = 7$ まで整数ステップで拡張可能ですが，計算上の制約から，実験では $\phi = 0$ と $\phi = 3$ のみを使用していますのでご注意ください．

4.1. Datasets

このサブセクションでは，2つの有名なベンチマークデータセットを用いて，我々のアプローチを評価します．

4.1.1 Linemod

Linemod[15]データセットは，6次元物体姿勢推定のためのベンチマークデータセットとして広く用いられている． Linemodデータセットは13個の異なる物体（実際には15個だが，他の多くの作品では13個しか使われていない[39][25][26][44][20][35]）からなり，13の散らかったシーンに配置されている．各シーンでは，他のオブジェクトが同時に見えているにもかかわらず，1つのオブジェクトだけが6Dポーズでアノテーションされています．そのため，我々のアプローチは，複数のオブジェクトを検出し，そのポーズを推定することができるにもかかわらず，各オブジェクトに対して1つのモデルをトレーニングする必要がありました．各オブジェクトには約1200のアノテーションが施されていますが，公平に比較するために，他の研究[3][26][39]と同じトレーニングとテストの分割を使用しています．この分割では，物体のポーズが 15° の最小角距離を持つように訓練画像を選択し，約15%の訓練画像と約85%のテスト画像を作成しました．また，合成された画像は一切使用していません．4.4節では，我々の結果を最先端の手法と比較する．

5 Conclusion

本論文では，最先端の2Dオブジェクト検出アーキテクチャの一種であるEfficientDet[38]をベースにした，拡張性の高いエンド・ツー・エンドの6Dオブジェクト・ポーズ推定アプローチであるEfficientPoseを紹介します． 直感的かつ効率的な方法でアーキテクチャを拡張し，ベースネットワークの利点を維持しつつ，追加の計算コストを低く抑えながら，2Dオブジェクト検出だけでなく，複数

のオブジェクトやインスタンスの6Dオブジェクトポーズ推定を，すべてシングルショットで実行します．我々のアプローチは，広く使用されているベンチマークデータセットLinemodにおいて，27 FPS以上でエンド・ツー・エンドで動作しながら，最先端の結果を達成しました．このように，6次元物体の姿勢を直接推定するホリスティックな手法は，同様の学習データ条件であれば，2D+PnP法と精度の面で競合することができます．さらに，2D+PnP法とは対照的に，我々の手法の実行時間はオブジェクトの数にほぼ依存しないため，ロボットによる把持や自律走行など，複数のオブジェクトが関与し，リアルタイム性に制約がある実世界のシナリオに適している．