**Abstract**

We present a conceptually simple, flexible, and general framework for object instance segmentation.

オブジェクトインスタンスのセグメンテーションのための概念的にシンプルで柔軟な一般的なフレームワークを示します。

Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance.

私たちのアプローチは、画像内のオブジェクトを効率的に検出すると同時に、インスタンスごとに高品質のセグメンテーションマスクを生成します。

The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.

Mask R-CNNと呼ばれるメソッドは、境界ボックス認識の既存のブランチと並行してオブジェクトマスクを予測するブランチを追加することにより、Faster R-CNNを拡張します。

Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework.

マスクR-CNNはトレーニングが簡単で、5 fpsで実行されるFaster R-CNNにわずかなオーバーヘッドしか追加しません。さらに、マスクR-CNNは他のタスクに簡単に一般化できます。たとえば、同じフレームワークで人間のポーズを推定できます。

We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, boundingbox object detection, and person keypoint detection.

インスタンスのセグメンテーション、境界ボックスオブジェクトの検出、人物のキーポイントの検出など、COCOの一連の課題の3つのトラックすべてで最高の結果を示しています。

Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners.

余計なものがない場合、Mask R-CNNは、COCO 2016チャレンジ受賞者を含む、すべてのタスクで既存のすべての単一モデルエントリよりも優れています。

We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition.

シンプルで効果的なアプローチが強固なベースラインとして機能し、インスタンスレベルの認識に関する今後の研究を容易にすることを願っています。

**1. Introduction**

The vision community has rapidly improved object detection and semantic segmentation results over a short period of time.

ビジョンコミュニティは、短時間でオブジェクト検出とセマンティックセグメンテーションの結果を急速に改善しました。

In large part, these advances have been driven by powerful baseline systems, such as the Fast/Faster RCNN [12, 36] and Fully Convolutional Network (FCN) [30] frameworks for object detection and semantic segmentation, respectively.

主に、これらの進歩は、それぞれオブジェクト検出およびセマンティックセグメンテーション用のFast / Faster RCNN [12、36]およびFully Convolutional Network（FCN）[30]フレームワークなどの強力なベースラインシステムによって推進されてきました。

These methods are conceptually intuitive and offer flexibility and robustness, together with fast training and inference time.

これらの方法は概念的に直感的であり、柔軟性と堅牢性を提供し、高速なトレーニングと推論時間を提供します。

Our goal in this work is to develop a comparably enabling framework for instance segmentation.

この作業における私たちの目標は、インスタンスのセグメンテーションのための比較的有効なフレームワークを開発することです。

Instance segmentation is challenging because it requires the correct detection of all objects in an image while also precisely segmenting each instance.

インスタンスのセグメンテーションは、各インスタンスを正確にセグメント化しながら、画像内のすべてのオブジェクトを正しく検出する必要があるため、困難です。

It therefore combines elements from the classical computer vision tasks of object detection, where the goal is to classify individual objects and localize each using a bounding box, and semantic segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.

したがって、オブジェクト検出の古典的なコンピュータービジョンタスクの要素を組み合わせます。目標は個々のオブジェクトを分類し、境界ボックスを使用してそれぞれをローカライズすることです。セマンティックセグメンテーションは目標を区別せずに各ピクセルを固定のカテゴリセットに分類することです。オブジェクトインスタンス。

Given this, one might expect a complex method is required to achieve good results. However, we show that a surprisingly simple, flexible, and fast system can surpass prior state-of-the-art instance segmentation results.

これを考えると、良い結果を得るには複雑な方法が必要になると予想されるかもしれません。ただし、驚くほどシンプルで、柔軟性があり、高速なシステムは、従来の最先端のインスタンスセグメンテーション結果を上回ることができることを示しています。

Our method, called Mask R-CNN, extends Faster R-CNN [36] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression (Figure 1).

Mask R-CNNと呼ばれるこの方法は、各関心領域（RoI）のセグメンテーションマスクを予測するためのブランチを、分類およびバウンディングボックス回帰の既存のブランチと並行して追加することで、より高速なR-CNN [36]を拡張します（図1） 。

The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-topixel manner.

マスクブランチは、各RoIに適用される小さなFCNであり、ピクセルごとにセグメンテーションマスクを予測します。

Mask R-CNN is simple to implement and train given the Faster R-CNN framework, which facilitates a wide range of flexible architecture designs.

Mask R-CNNは、より柔軟なアーキテクチャ設計を容易にするFaster R-CNNフレームワークを前提として、実装とトレーニングが簡単です。

Additionally, the mask branch only adds a small computational overhead, enabling a fast system and rapid experimentation.

さらに、マスクブランチはわずかな計算オーバーヘッドを追加するだけで、高速システムと迅速な実験を可能にします。

In principle Mask R-CNN is an intuitive extension of Faster R-CNN, yet constructing the mask branch properly is critical for good results. Most importantly, Faster RCNN was not designed for pixel-to-pixel alignment between network inputs and outputs.

原則として、Mask R-CNNはFaster R-CNNの直感的な拡張ですが、良好な結果を得るにはマスクブランチを適切に構築することが重要です。最も重要なことは、Faster RCNNは、ネットワークの入力と出力の間のピクセル間のアライメントのために設計されていないことです。

This is most evident in how RoIPool [18, 12], the de facto core operation for attending to instances, performs coarse spatial quantization for feature extraction.

これは、インスタンスに対応するための事実上のコア操作であるRoIPool [18、12]が、特徴抽出のために粗い空間量子化を実行する方法で最も明白です。

To fix the misalignment, we propose a simple, quantization-free layer, called RoIAlign, that

faithfully preserves exact spatial locations.

ミスアライメントを修正するために、正確な空間位置を忠実に保持するRoIAlignと呼ばれる、量子化のないシンプルなレイヤーを提案します。

Despite being a seemingly minor change, RoIAlign has a large impact: itimproves mask accuracy by relative 10% to 50%, showing bigger gains under stricter localization metrics.

一見小さな変更であるにもかかわらず、RoIAlignは大きな影響を及ぼします。それは、マスクの精度を10％〜50％改善し、より厳密なローカリゼーションメトリックの下でより大きなゲインを示します。

Second, we found it essential to decouple mask and class prediction: we predict a binary mask for each class independently, without competition among classes, and rely on the network’s RoI classification branch to predict the category.

次に、マスクとクラスの予測を分離することが不可欠であることがわかりました。クラス間の競合なしに各クラスのバイナリマスクを独立して予測し、ネットワークのRoI分類ブランチに依存してカテゴリを予測します。

In contrast, FCNs usually perform per-pixel multi-class categorization, which couples segmentation and classification, and based on our experiments works poorly for instance segmentation.

対照的に、FCNは通常、ピクセルごとのマルチクラス分類を実行します。これは、セグメンテーションと分類を結合し、実験に基づいてインスタンスのセグメンテーションではうまく機能しません。

Without bells and whistles, Mask R-CNN surpasses all previous state-of-the-art single-model results on the COCO instance segmentation task [28], including the heavilyengineered entries from the 2016 competition winner.

余計なことをすることなく、Mask R-CNNは、2016年のコンペティション優勝者からの厳しく設計されたエントリーを含む、COCOインスタンスのセグメンテーションタスク[28]での以前のすべての最先端の単一モデルの結果を凌surします。

As a by-product, our method also excels on the COCO object detection task.

副産物として、この方法はCOCOオブジェクト検出タスクにも優れています。

In ablation experiments, we evaluate multiple basic instantiations, which allows us to demonstrate its robustness and analyze the effects of core factors.

アブレーション実験では、複数の基本的なインスタンス化を評価します。これにより、その堅牢性を実証し、コアファクターの効果を分析できます。

Our models can run at about 200ms per frame on a GPU, and training on COCO takes one to two days on a single 8-GPU machine.

私たちのモデルは、GPUで1フレームあたり約200msで実行でき、COCOのトレーニングは、単一の8-GPUマシンで1〜2日かかります。

We believe the fast train and test speeds, together with the framework’s flexibility and accuracy, will benefit and ease future research on instance segmentation.

フレームワークの柔軟性と正確さとともに、高速な列車とテストの速度は、インスタンスのセグメンテーションに関する将来の研究に役立つと信じています。

Finally, we showcase the generality of our framework via the task of human pose estimation on the COCO keypoint dataset [28].

最後に、COCOキーポイントデータセットに対する人間の姿勢推定のタスクを介して、フレームワークの一般性を示します[28]。

By viewing each keypoint as a one-hot binary mask, with minimal modification Mask R-CNN can be applied to detect instance-specific poses.

各キーポイントをワンホットバイナリマスクとして表示することにより、最小限の変更でマスクR-CNNを適用してインスタンス固有のポーズを検出できます。

Mask R-CNN surpasses the winner of the 2016 COCO keypoint competition, and at the same time runs at 5 fps. Mask R-CNN, therefore, can be seen more broadly as a flexible framework

for instance-level recognition and can be readily extended to more complex tasks.

マスクR-CNNは、2016 COCOキーポイントコンペティションの勝者を上回り、同時に5 fpsで実行されます。したがって、マスクR-CNNは、より広く、柔軟なフレームワークとして見ることができます。インスタンスレベルの認識など、より複雑なタスクに容易に拡張できます。

We have released code to facilitate future research.

将来の研究を促進するためのコードをリリースしました。

**2. Related Work**

R-CNN: The Region-based CNN (R-CNN) approach [13] to bounding-box object detection is to attend to a manageable number of candidate object regions [42, 20] and evaluate convolutional networks [25, 24] independently on each RoI.

R-CNN：境界ボックスオブジェクト検出に対する領域ベースのCNN（R-CNN）アプローチ[13]は、管理可能な数の候補オブジェクト領域[42、20]に対応し、畳み込みネットワーク[25、24]を個別に評価することです。各RoIで。

R-CNN was extended [18, 12] to allow attending to RoIs on feature maps using RoIPool, leading to fast speed and better accuracy.

R-CNNは[18、12]に拡張され、RoIPoolを使用して機能マップでRoIに参加できるようになり、高速で精度が向上しました。

Faster R-CNN [36] advanced this stream by learning the attention mechanism with a Region Proposal Network (RPN).

Faster R-CNN [36]は、Region Proposal Network（RPN）を使用してアテンションメカニズムを学習することにより、このストリームを進めました。

Faster R-CNN is flexible and robust to many follow-up improvements (e.g., [38, 27, 21]), and is the current leading framework in several benchmarks.

より高速なR-CNNは、多くのフォローアップの改善（たとえば[38、27、21]）に対して柔軟で堅牢であり、いくつかのベンチマークにおける現在の主要なフレームワークです。

Instance Segmentation: Driven by the effectiveness of RCNN, many approaches to instance segmentation are based on segment proposals.

インスタンスのセグメンテーション：RCNNの有効性により、インスタンスのセグメンテーションへの多くのアプローチは、セグメントの提案に基づいています。

Earlier methods [13, 15, 16, 9] resorted to bottom-up segments [42, 2].

以前の方法[13、15、16、9]は、ボトムアップセグメント[42、2]に頼っていました。

DeepMask [33] and following works [34, 8] learn to propose segment candidates, which are then classified by Fast R-CNN.

DeepMask [33]および次の作品[34、8]は、セグメント候補の提案方法を学習し、Fast R-CNNによって分類されます。

In these methods, segmentation precedes recognition, which is slow and less accurate.

これらの方法では、セグメンテーションが認識に先行しますが、認識は遅く、精度が低下します。

Likewise, Dai et al. [10] proposed a complex multiple-stage cascade that predicts segment proposals from bounding-box proposals, followed by classification.

同様に、Dai et al。 [10]は、バウンディングボックスの提案からセグメントの提案を予測し、その後に分類を行う複雑な多段階カスケードを提案しました。

Instead, our method is based on parallel prediction of masks and class labels, which is simpler and more flexible.

代わりに、このメソッドは、マスクとクラスラベルの並列予測に基づいています。これは、よりシンプルで柔軟です。

Most recently, Li et al. [26] combined the segment proposal system in [8] and object detection system in [11] for “fully convolutional instance segmentation” (FCIS).

最近では、李ら。 [26] [8]のセグメント提案システムと[11]のオブジェクト検出システムを組み合わせて、「完全畳み込みインスタンスセグメンテーション」（FCIS）を実現しました。

The common idea in [8, 11, 26] is to predict a set of positionsensitive output channels fully convolutionally.

[8、11、26]の一般的な考え方は、位置に敏感な出力チャネルのセットを完全に畳み込みで予測することです。

These channels simultaneously address object classes, boxes, and masks, making the system fast.

これらのチャネルは、オブジェクトクラス、ボックス、およびマスクに同時に対処し、システムを高速化します。

But FCIS exhibits systematic errors on overlapping instances and creates spurious edges (Figure 6), showing that it is challenged by the fundamental difficulties of segmenting instances.

しかし、FCISは重複するインスタンスで系統的なエラーを示し、スプリアスエッジを作成し（図6）、インスタンスのセグメント化の根本的な困難に挑戦されていることを示しています。

Another family of solutions [23, 4, 3, 29] to instance segmentation are driven by the success of semantic segmentation.

インスタンスセグメンテーションに対するソリューションの別のファミリ[23、4、3、29]は、セマンティックセグメンテーションの成功によって推進されます。

Starting from per-pixel classification results (e.g., FCN outputs), these methods attempt to cut the pixels of the same category into different instances.

ピクセルごとの分類結果（FCN出力など）から開始すると、これらのメソッドは同じカテゴリのピクセルを異なるインスタンスにカットしようとします。

In contrast to the segmentation-first strategy of these methods, Mask R-CNN is based on an instance-first strategy.

これらのメソッドのセグメンテーション優先戦略とは対照的に、マスクR-CNNはインスタンス優先戦略に基づいています。

We expect a deeper incorporation of both strategies will be studied in the future.

将来的には、両方の戦略をより深く取り入れることが検討されると予想しています。

**3. Mask R-CNN**

Mask R-CNN is conceptually simple: Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; to this we add a third branch that outputs the object mask.

マスクR-CNNは概念的にシンプルです。高速R-CNNには、各候補オブジェクトに対して2つの出力、クラスラベルおよび境界ボックスオフセットがあります。これに、オブジェクトマスクを出力する3番目のブランチを追加します。

Mask R-CNN is thus a natural and intuitive idea.

したがって、マスクR-CNNは自然で直感的なアイデアです。

But the additional mask output is distinct from the class and box outputs, requiring extraction of much finer spatial layout of an object.

ただし、追加のマスク出力はクラスおよびボックス出力とは異なるため、オブジェクトの非常に細かい空間レイアウトを抽出する必要があります。

Next, we introduce the key elements of Mask R-CNN, including pixel-to-pixel alignment, which is the main missing piece of Fast/Faster R-CNN.

次に、Fast / Faster R-CNNの主要な欠落部分であるピクセル間のアライメントを含む、Mask R-CNNの主要な要素を紹介します。

Faster R-CNN: We begin by briefly reviewing the Faster R-CNN detector [36].

Faster R-CNN：Faster R-CNN検出器を簡単にレビューすることから始めます[36]。

Faster R-CNN consists of two stages.

より高速なR-CNNは2つのステージで構成されます。

The first stage, called a Region Proposal Network (RPN), proposes candidate object bounding boxes.

Region Proposal Network（RPN）と呼ばれる最初の段階では、候補オブジェクトの境界ボックスを提案します。

The second stage, which is in essence Fast R-CNN [12], extracts features using RoIPool from each candidate box and performs classification and bounding-box regression.

本質的にはFast R-CNN [12]である2番目の段階では、各候補ボックスからRoIPoolを使用して特徴を抽出し、分類および境界ボックス回帰を実行します。

The features used by both stages can be shared for faster inference.

両方のステージで使用される機能を共有して、推論を高速化できます。

We refer readers to [21] for latest, comprehensive comparisons between Faster R-CNN and other frameworks.

Faster R-CNNと他のフレームワークの最新の包括的な比較については、読者に[21]を参照してください。

Mask R-CNN: Mask R-CNN adopts the same two-stage procedure, with an identical first stage (which is RPN).

マスクR-CNN：マスクR-CNNは、同じ2段階の手順を採用します。最初の段階（RPN）は同じです。

In the second stage, in parallel to predicting the class and box offset, Mask R-CNN also outputs a binary mask for each RoI.

第2段階では、クラスとボックスオフセットの予測と並行して、Mask R-CNNは各RoIのバイナリマスクも出力します。

This is in contrast to most recent systems, where classification depends on mask predictions (e.g. [33, 10, 26]).

これは、分類がマスク予測に依存する最新のシステムとは対照的です（例[33、10、26]）。

Our approach follows the spirit of Fast R-CNN [12] that applies bounding-box classification and regression in parallel (which turned out to largely simplify the multi-stage pipeline of original R-CNN [13]).

私たちのアプローチは、境界ボックス分類と回帰を並行して適用する高速R-CNN [12]の精神に従います（元のR-CNN [13]のマルチステージパイプラインを大幅に簡素化することが判明しました）。

Formally, during training, we define a multi-task loss on each sampled RoI as L = Lcls + Lbox + Lmask.

正式には、トレーニング中に、サンプリングされた各RoIでのマルチタスク損失をL = Lcls + Lbox + Lmaskとして定義します。

The classification loss Lcls and bounding-box loss Lbox are identicalas those defined in [12].

分類損失Lclと境界ボックス損失Lboxは、[12]で定義されているものと同一です。

The mask branch has a Km2-dimensional output for each RoI, which encodes K binary masks of resolution , one for each of the K classes.

マスクブランチには、各RoIに対してKm2次元の出力があり、解像度m×mのK個のバイナリマスク（Kクラスごとに1つ）をエンコードします。

To this we apply a per-pixel sigmoid, and define Lmask as the average binary cross-entropy loss. For an RoI associated with ground-truth class k, Lmask is only defined on the k-th mask (other mask outputs do not contribute to the loss).

これにピクセル単位のシグモイドを適用し、Lmaskを平均バイナリクロスエントロピー損失として定義します。グラウンドトゥルースクラスkに関連付けられたRoIの場合、Lmaskはk番目のマスクでのみ定義されます（他のマスク出力は損失に寄与しません）。

Our definition of Lmask allows the network to generate masks for every class without competition among classes; we rely on the dedicated classification branch to predict the class label used to select the output mask. This decouples mask and class prediction.

Lmaskの定義により、ネットワークはクラス間の競合なしにすべてのクラスのマスクを生成できます。専用の分類ブランチに依存して、出力マスクの選択に使用されるクラスラベルを予測します。これにより、マスクとクラスの予測が分離されます。

This is different from common practice when applying FCNs [30] to semantic segmentation, which typically uses a per-pixel softmax and a multinomial cross-entropy loss.

これは、セマンティックセグメンテーションにFCN [30]を適用する際の一般的な手法とは異なります。セマンティックセグメンテーションでは、通常、ピクセルごとのソフトマックスと多項クロスエントロピー損失が使用されます。

In that case, masks across classes compete; in our case, with a per-pixel sigmoid and a binary loss, they do not.

その場合、クラス全体でマスクが競合します。私たちの場合、ピクセルごとのシグモイドとバイナリ損失では、そうではありません。

We show by experiments that this formulation is key for good instance segmentation results.

この定式化は、良好なインスタンスのセグメンテーション結果の鍵であることを実験により示しています。

Mask Representation: A mask encodes an input object’s spatial layout.

マスク表現：マスクは、入力オブジェクトの空間レイアウトをエンコードします。

Thus, unlike class labels or box offsets that are inevitably collapsed into short output vectors by fully-connected (fc) layers, extracting the spatial structure of masks can be addressed naturally by the pixel-to-pixel correspondence provided by convolutions.

したがって、完全に接続された（fc）レイヤーによって必然的に短い出力ベクトルに折りたたまれるクラスラベルまたはボックスオフセットとは異なり、マスクの空間構造の抽出は、畳み込みによって提供されるピクセル間の対応によって自然に対処できます。

Specifically, we predict an mask from each RoI using an FCN [30].

具体的には、FCNを使用して各RoIからm×mマスクを予測します[30]。

This allows each layer in the mask branch to maintain the explicit object spatial layout without collapsing it into a vector representation that lacks spatial dimensions.

これにより、マスクブランチの各レイヤーは、空間次元のないベクトル表現に折りたたむことなく、明示的なm×mオブジェクトの空間レイアウトを維持できます。

Unlike previous methods that resort to fc layers for mask prediction [33, 34, 10], our fully convolutional representation requires fewer parameters, and is more accurate as demonstrated by experiments.

マスク予測のためにfcレイヤーに頼る以前の方法[33、34、10]とは異なり、完全な畳み込み表現では必要なパラメーターが少なく、実験で実証されているようにより正確です。

This pixel-to-pixel behavior requires our RoI features, which themselves are small feature maps, to be well aligned to faithfully preserve the explicit per-pixel spatial correspondence.

このピクセル間動作では、明示的なピクセルごとの空間的対応を忠実に維持するために、RoI機能自体が小さな機能マップである必要があります。

This motivated us to develop the following RoIAlign layer that plays a key role in mask prediction.

これにより、マスク予測で重要な役割を果たす次のRoIAlignレイヤーを開発することになりました。

RoIAlign: RoIPool [12] is a standard operation for extracting a small feature map (e.g., ) from each RoI.

RoIAlign：RoIPool [12]は、各RoIから小さな機能マップ（7x7など）を抽出するための標準操作です。

RoIPool first quantizes a floating-number RoI to the discrete granularity of the feature map, this quantized RoI is then subdivided into spatial bins which are themselves quantized, and finally feature values covered by each bin are aggregated (usually by max pooling).

RoIPoolは最初に浮動小数点RoIを特徴マップの離散粒度に量子化し、この量子化されたRoIはそれ自体が量子化される空間ビンに再分割され、最後に各ビンによってカバーされる特徴値が集約されます（通常は最大プーリングによって）。

Quantization is performed, e.g., on a continuous coordinate x by computing [x=16], where 16 is a feature map stride and [] is rounding; likewise, quantization is performed when dividing into bins (e.g., ).

量子化は、たとえば[x = 16]を計算することにより、連続座標xで実行されます。ここで、16は機能マップストライド、[∙]は丸めです。同様に、ビンに分割するときに量子化が実行されます（7×7など）。

These quantizations introduce misalignments between the RoI and the extracted features.

While this may not impact classification, which is robust to small translations, it has a large negative effect on predicting pixel-accurate masks.

これらの量子化により、RoIと抽出された特徴の間に不整合が生じます。

これは、小さな翻訳にはロバストな分類には影響しないかもしれませんが、ピクセル精度のマスクの予測に大きな悪影響を及ぼします。

To address this, we propose an RoIAlign layer that removes the harsh quantization of RoIPool, properly aligning the extracted features with the input. Our proposed change is simple: we avoid any quantization of the RoI boundaries or bins (i.e., we use x=16 instead of [x=16]).

これに対処するために、RoIPoolの過酷な量子化を除去し、抽出されたフィーチャを入力に適切に位置合わせするRoIAlignレイヤーを提案します。提案された変更は簡単です。RoIの境界またはビンの量子化を避けます（つまり、[x = 16]の代わりにx = 16を使用します）。

We use bilinear interpolation [22] to compute the exact values of the input features at four regularly sampled locations in each RoI bin, and aggregate the result (using max or average), see Figure 3 for details.

バイリニア補間[22]を使用して、各RoIビンの4つの定期的にサンプリングされた位置で入力フィーチャの正確な値を計算し、結果を集計します（最大または平均を使用）。詳細については図3を参照してください。

We note that the results are not sensitive to the exact sampling locations, or how many points are sampled, as long as no quantization is performed.

結果は、量子化が実行されない限り、正確なサンプリング位置やサンプリングされたポイント数に影響されないことに注意してください。

RoIAlign leads to large improvements as we show in x4.2.

RoIAlignは、x4.2で示したように大幅に改善されています。

We also compare to the RoIWarp operation proposed in [10].

また、[10]で提案されているRoIWarp操作と比較します。

Unlike RoIAlign, RoIWarp overlooked the alignment issue and was implemented in [10] as quantizing RoI just like RoIPool.

RoIAlignとは異なり、RoIWarpはアライメントの問題を見落とし、RoIPoolのようにRoIを量子化するように[10]で実装されました。

So even though RoIWarp also adopts bilinear resampling motivated by [22], it performs on par with RoIPool as shown by experiments (more details in Table 2c), demonstrating the crucial role of alignment.

そのため、RoIWarpは[22]によって動機付けられたバイリニアリサンプリングも採用していますが、実験（表2cに詳細を示す）で示すようにRoIPoolと同等のパフォーマンスを発揮し、アライメントの重要な役割を実証します。