

[ディープラーニング活用事例と使いこなしの勘所] 画像処理分野

1 ニュース記事画像の自動クロッピング 応 専

田中智大 | ヤフー (株)

画像のクロッピング

スマートフォン向けの Yahoo! JAPAN アプリでは、**図-1**のようにタイムライン形式でニュース記事が配信される。ここで表示される画像は、正方形や横長の長方形などさまざまなアスペクト比を取り得る。しかし記事に付随する画像は必ずしも要求するアスペクト比ではないため、適切な形に加工する処理が必要になる。画像の形状加工にはリサイズや余白の追加などのさまざまな方法があるが、Yahoo! JAPAN アプリでは画像の一部を切り出して、表示したいアスペクト比にするクロッピング処理を採用している。また、ニュース記事の例ではないが、Twitter のタイムライン上に表示される

画像に対してもクロッピング処理が行われているという報告がある¹⁾。

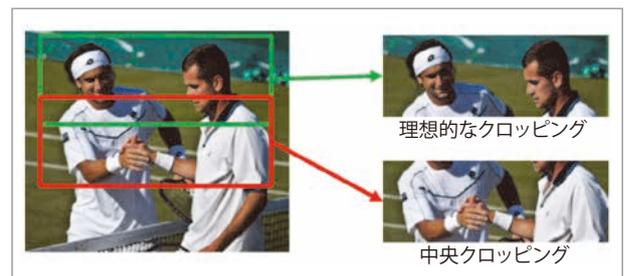
クロッピングの自動化と課題

ニュース記事は日々大量に入稿されるため、画像のクロッピング処理は人手を介さずに自動で行えるのが望ましい。しかし、単純に画像の中央からクロッピング処理を行うと、**図-2**のように画像中の重要な物体（このケースでは人の顔）が欠損してしまうことがある。そのため、画像中の重要な領域を解析し、それらを欠損させないクロッピング処理を実現する必要がある。

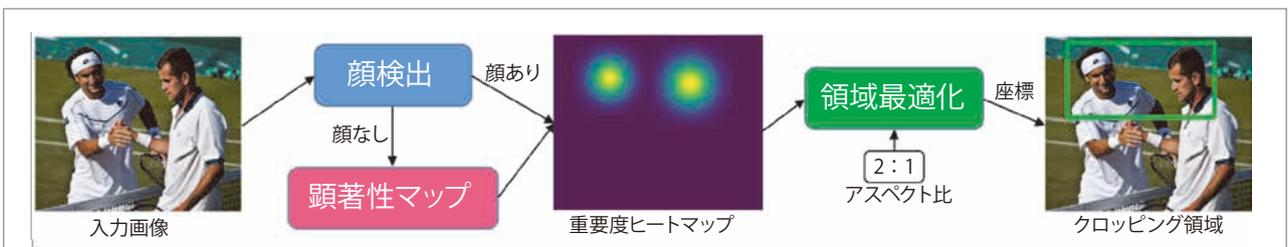
図-3に実際の自動クロッピング処理の概略図を示す。ニュース記事の画像では人の顔が重要になることが多いため、まず顔検出処理を行う。顔が検出された場合は、検出された顔の領域を基に重要な領域を表すヒートマップを作成する。顔が検出されなかった場合は画



■ 図-1 Yahoo! JAPAN アプリ



■ 図-2 中央クロッピングの問題



■ 図-3 ニュース記事画像の自動クロッピング処理の概略図



像から顕著性マップを推定し、ヒートマップとして利用する。顕著性マップとは、色情報などの画像特徴量から、画像中の注目されやすい領域を視覚化したものである。最後に、ヒートマップと入力したアスペクト比に加え、撮影者が被写体を中央に入れようとする傾向を考慮して、重要度の高いところの割合が大きくなるようにクロッピングする領域を最適化する。

アプリのリリース当初は、クロッピング処理内の顔検出と顕著性マップの推定を OpenCV^{☆1} などに実装されている古典的な手法や、画像の色情報などの単純な特徴量に基づいた手法で行っていた。そのため顔の誤検出等が原因のクロッピングミスがたびたび発生していた。そこで 2015 年頃より、当時盛り上がり始めていたディープラーニングを用いた画像の自動クロッピング手法の検討を開始した。

ディープラーニングの導入

どこに導入するか

まず、ディープラーニングをクロッピング処理のどの部分に導入するかを検討する必要がある。理想的には、画像とアスペクト比を入力すると最適なクロッピング領域が得られるように、クロッピング処理全体をディープラーニングでモデリングできるとよい。しかし、教師データの問題がある。原則としてディープラーニングモデルの学習には大量に教師データが必要なため、教師データを多数用意できない場合は高い性能が期待できない。また、クロッピングは主観的要素を含むため教師データを作成する人の個人差が強く影響する。そのため、同一画像・同一アスペクト比でも正解が一意に決まらず教師データを作成するのが難しい。以上の理由から、今回は図-3に示すクロッピング処理を踏襲し、処理内の顔検出と顕著性マップ推定にそれぞれディープラーニングを導入することとした。

モデリング

ディープラーニングモデルを作成するにあたって、使

用するフレームワーク、モデル・手法、教師データ等を検討する必要がある。

ディープラーニングのフレームワークは、今ではさまざまなものが公開されているが、検討を始めた 2015 年時点では Caffe^{☆2} というフレームワークが画像の分野では一般的に使われており、これを採用した。

顔検出については、当時性能が良かった Cascade CNN²⁾ を採用し、Caffe を使って実装した。学習に使う教師データは、幸いにも社内でデータ作成の人的リソースが確保できたため、約 3 万枚の画像に対して顔領域の座標情報を付与したものを作成した。

顕著性マップ推定については、画像分類用のディープラーニングモデルを応用した手法³⁾ を独自に考案・実装した。ここでは、ImageNet と呼ばれる巨大な公開データセットで学習済みの画像分類モデルを利用した。ImageNet を使った画像分類モデルは、Model Zoo^{☆3} という学習済みモデル共有サイトに公開されているものがある。ライセンスに問題がなければそれらのモデルをそのまま利用することができる。

導入の効果

まず、顔検出と顕著性マップ推定モデルの性能評価を個別に行った。顔検出については教師データを十分に用意できたこともあり、従来のモデルと比較して検出できる顔の数が約 30% 増加し、顔の誤検出数は約 30% 削減できた。一方、顕著性マップ推定については従来の手法と比べて性能向上があまり大きくなかった。そのため、精度と処理速度・メモリ使用量などを総合的に考慮して、従来の手法を引き続き利用することとした。

次にクロッピングの処理性能の面から、ディープラーニングを用いた顔検出処理の導入効果を評価した。本件では評価方法として、画像約 1,000 枚に対して、従来手法とディープラーニングを導入した手法を使ってそれぞれクロッピングし、評価者 4 名にどちらの結果が良いかを 4 段階で評価してもらった。この結果を図-4

☆1 画像処理のオープンソースライブラリ。

☆2 <http://caffe.berkeleyvision.org/>

☆3 <https://github.com/BVLC/caffe/wiki/Model-Zoo>

に示す。この結果から、顔検出をディープラーニングモデルに置き換えたことで、クロッピング処理の性能が改善していることが分かる。これは顔の誤検出によるクロッピングミスが減ったことが主な理由と考えられる。一方で従来版の方が良かった画像も20%あることが分かる。これは検出できる顔の数が増えたことで、画像中のあまり重要ではない人物の顔も検出してしまい、逆にクロッピング結果に悪影響を及ぼしていることが原因として考えられる。最後に、具体的なクロッピングの改善例を図-5に示す。従来版では顔が検出できず、顕著性マップを利用しているが、ほぼ画像の中央をクロッピングしてしまっている。一方でディープラーニングを導入した手法では、検出された顔領域を中心とした領域でクロッピングできている。

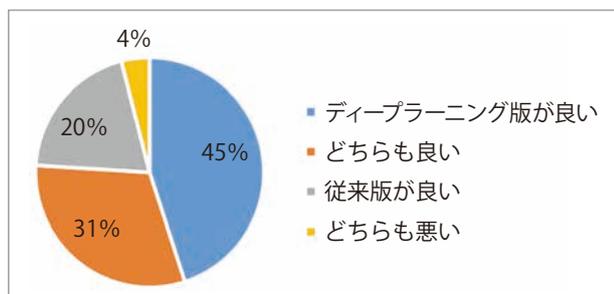
ディープラーニング導入のポイント

導入時の注意点

今回は検討の結果、最終的にクロッピング処理の一部である顔検出処理だけをディープラーニングを使ったモデルに置き換えた。ディープラーニングは非常に強力なツールであるが、教師データ量や、解決したい問題によっては必ずしも十分な性能を発揮できるわけではない。そういったケースではディープラーニングをすべてに導入するのではなく、問題を切り分け、部分的にディープラーニングが有効に働く処理がないかを検討してみるとよい。

実運用時における課題

最近では、汎用的な物体検出モデルが次々に考案さ



■図-4 クロッピング処理の評価結果

れている。実装が公開されることも多いため、モデリングは当時より随分容易になっている。しかし実際のサービス要件に応じて、作ったモデルを導入・運用するという点についてはいまだに難しいと感じる。今回作成した顔検出モデルは、従来のモデルに比べて処理速度が遅くなってしまったため、そのままではサービスに導入するのが難しかった。当時はディープラーニングモデルの高速化のノウハウも少なかったため、最終的にプロダクション環境にGPUサーバを導入することで速度の問題を解決した。しかし、一般的にGPUサーバは高価であり、CPUのみのサーバと比べて運用コストも高い。そのため学習したモデルを動かすだけであればCPUだけで高速に処理できるのが望ましい。最近では、CPUでも比較的高速に動作するフレームワークが増えてきており、たとえばCaffeにおいてもIntel製のCPU向けにチューニングされた実装が公開されている^{☆4}。CPU上での処理速度が問題になるケースではこのようなフレームワークを利用することで高速化でき導入が容易になる可能性もある。

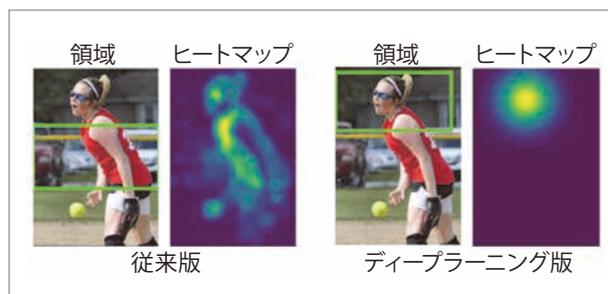
参考文献

- 1) Theis, L. and Wang, Z. : Speedy Neural Networks for Smart Auto-Cropping of Images, https://blog.twitter.com/engineering/en_us/topics/infrastructure/2018/Smart-Auto-Cropping-of-Images.html (2018年7月18日アクセス)
- 2) Haoxiang, L. et al. : A Convolutional Neural Network Cascade for Face Detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015).
- 3) 山下直晃, 田中智大 : A Simple Method to Create Enhanced Saliency Map with CNN Classifier, 画像の認識・理解シンポジウム (2016). (2018年7月30日受付)

☆4 <https://github.com/intel/caffe>

■田中智大 tomotana@yahoo-corp.jp

2011年電気通信大学修士。同年ヤフー(株)入社。エンジニアとして画像処理技術のサービス活用に従事。



■図-5 クロッピングの改善例