

DeepFoodCam: DCNN による 101 種類食事認識アプリ

岡元 晃一¹ 柳井 啓司¹

1. はじめに

本デモでは、ディープラーニングを用いたスマートフォンアプリケーション DeepFoodCam の実演を行う。我々は、これまでに、BoF や Fisher Vector を用いたリアルタイム食事認識アプリ FoodCam [2] を構築している。DeepFoodCam では、認識手法を Deep Convolutional Neural Network (DCNN) に変更し、大幅な認識精度向上を実現した。

2. 大規模 DCNN による認識アプリ

DCNN による物体認識スマートフォンアプリはトロント大によるもの^{*1} が現在公開されているが、これはサーバベースの認識であり、アプリ自体は単に画像をサーバに送信し、結果を受け取って表示するだけの機能しかない。クライアントアプリ側で DCNN で一般物体認識を行うアプリはかつては JetPac spotter という iOS アプリが公開されていたが、現在は JetPac 社が Google によって買収されてしまったため、公開されていない。ただし、その認識エンジンの DeepBeliefSDK^{*2} はオープンソースとして公開されており、これを用いることで、iOS や Android で誰もが簡単に AlexNet をベースとした認識エンジンをアプリに組み込むことができる。ただし、AlexNet を使っているために 6000 万個の学習済パラメータが必要であり、DCNN パラメータの分布の範囲が狭い性質を利用して、1 パラメータ 1byte で表現する工夫をしているものの 60MB の model ファイルを保持する必要があるため、アプリサイズは大きなものとなってしまった。なお、アムステルダム大のベンチャーである EU vision も ILSVRC2013 の開催前後に AlexNet をベースとした 1000 種類認識 iOS アプリを一時的に公開していたが、こちらはパラメータ圧縮の工夫をしていなかったためアプリサイズが 250MB を超える巨大なものとなっていた。ちなみに、EU Vision は現在 Qualcomm に買収されている。

これに対して、DeepFoodCam では DCNN には一般的な AlexNet [4] ではなく、全結合層を持たないためパラメータ数が大幅に少ない Network-In-Network (NIN) [6] を用いる。AlexNet は 6000 万パラメータが必要なのに対して、NIN は 750 万パラメータで同程度の認識性能を 1000 種類認識において実現している。

3. 学習

DCNN は 4 つの畳み込み層を持った NIN のモデルを利用



図 1 システム画面と認識例。

用した。学習には Caffe^{*3}を利用して、最初にスクラッチから ILSVRC1000 種類と食事関連した 1000 種類の ImageNet 画像の 2000 クラス 210 万枚で学習し、そのモデルを UEC-FOOD100^{*4}の食事 100 クラスと、主に Twitter から収集した UECFOOD100 種類の食事名がメッセージに含まれていたにもかかわらず食事写真ではなかった食事写真と間違えやすい非食事画像 1 万枚の合計 101 クラスでファインチューニングした。

4. 実装

大規模一般物体認識用の DCNN をモバイルに実装する上で問題となるのは、メモリと実行速度である。

省メモリ化のために、750 万パラメータで済む network-in-network [6] 採用し、さらにパラメータ圧縮を行った。1 パラメータあたり 4bit に圧縮することによって、わずか 3.8MB で学習パラメータを保持することが可能となった。これは、従来の Fisher Vector と SVM 重みの圧縮を用いた場合 [3] のパラメータサイズ 9MB よりも大幅に小さいサイズとなっている。

一方、高速化には、(1) マルチコア対応、(2) スマートフォンで使われている ARM プロセッサの SIMD 命令 Neon の利用、(3) 計算時間の掛る畳み込み層の計算の簡略化、が挙げられる。

原稿執筆時点で、(1) は対応済みであり、それ以外は実装中である。図 1 に原稿執筆時点での DeepFoodCam のシステム画面と認識例を示す。なお、スマートフォン用の認識エンジンは、Caffe のモデルファイルおよび学習済パラメータから認識用の C 言語コードを自動生成するジェネレータを実装することによって実現した。

(1) だけの工夫で、クアッドコアスマートフォン Nexus5 上で 1 枚の認識に約 2.5 秒掛かっている。さらに (2)(3) を行い、実際のデモ時までには 1 秒未満を実現する予定である。なお、DeepBeliefSDK では iPhone5s 上で 300ms で実現出来ていると SDK のドキュメントに書かれており、(1)(2) の工夫が行われているようである。なお、スマート

¹ 電気通信大学大学院 情報理工学専攻 総合情報学専攻

^{*1} <https://play.google.com/store/apps/details?id=utoronto.deeplearning>

^{*2} <https://github.com/jetpacapp/DeepBeliefSDK>

^{*3} <http://caffe.berkeleyvision.org/>

^{*4} <http://foodcam.mobi/dataset/>

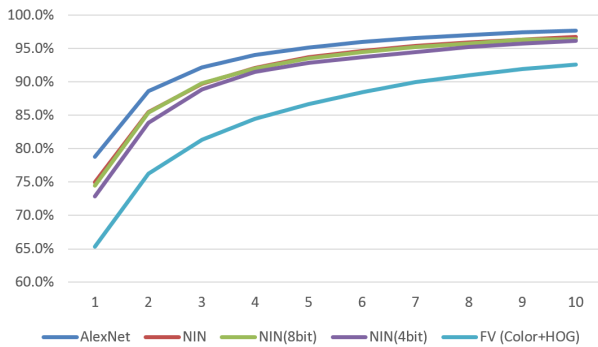


図 2 100 種類食事分類における 1 位から 10 位までの認識率。

フォンのプロセッサユニットに GPU が内蔵されている場合、それを利用することも考えられるが、動作環境が限定されるので、現時点では検討していない。

(3) については現在、世界中で多くの研究が行われている。DeepFoodCam では、畳み込みのフィルタの学習済パラメータのテンソルを低ランク分解 (CP 分解) をして、さらにファインチューニングする Fine-tuned CP-Decomposition [5] か、最初から畳み込み層を分解してから学習する Flattened CNN [1] を採用することを検討している。どちらも、分解してからファインチューニングすることによって、認識性能はほとんど低下しないことが示されている。

5. 認識例と評価結果

認識例を図 3 に示す。1 位で正解でない場合でも、5 位以内にはほんどの場合正解が含まれていることが分かる。図 3(f) が 1 位は寿司であったが、2 位は非食事であった。

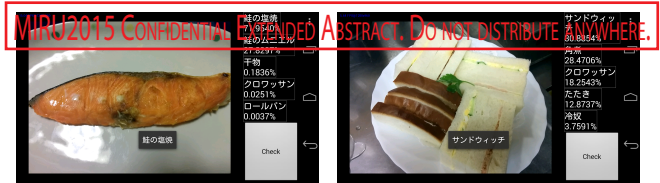
非食事を除いた 100 クラス分類の評価を 5 fold cross validation で行った。結果を図 2 に示す。パラメータ圧縮なしの場合、認識率は 1 位で 75.0%, 5 位以内で正解は 93.7% であった。1 パラメータあたり 8bit, 4bit にそれぞれ圧縮すると、必要なメモリ量は 30MB から 7.5MB, 3.8MB に減少し、認識性能は 1 位で 74.5%, 72.9%, 5 位以内で 93.5%, 92.9% であった。実験の結果、8bit の場合は性能低下はほとんど見られなかったため、デモ用のアプリでは 8bit を採用したが、メモリ制約の厳しい場合は 4bit でも十分である。同様の学習を AlexNet をベースとして拡張した 8000 万パラメータからなるモデルを使って行った場合、1 位で正解は 78.7%, 5 位以内で正解は 95.2% であった。グラフ中の FV(Color+HOG) が従来手法の結果である。1 位、5 位以内でそれぞれ 9.7 ポイント、7.0 ポイントの性能向上を達成した。

また、食事/非食事の分類精度は、NIN が 98.5%, AlexNet の拡張版が 99.0% であった。ほぼ非食事写真を排除することが可能となっている。なお、この拡張 AlexNet による食事認識エンジンを用いた食事認識認識 TwitterBot を @foodimg_bot として公開している。写真を付けたツイートを @foodimg_bot 宛てに送ると、認識結果 5 位までの食事名とカロリーを自動返信する。

6. おわりに

本研究では、パラメータ数が大幅に少ない Network-In-Network(NIN) [6] を用いて DCNN による食事認識モバイルアプリ DeepFoodCam を実現した。実験により、一般的な AlexNet に遜色ないことが示された。

今後はさらなる高速化と高精度化を予定している。また、スマートフォン上で食事を 3 次元的に分析して正確な



(a) 成功例 (鮭の塩焼)

(b) 成功例 (サンドウィッチ)



(c) 成功例 (非食事画像)



(d) 失敗例 (サラダは 2 位)



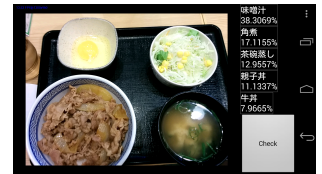
(e) 失敗例 (5 位以内にホットドッグなし)



(f) 失敗例 (1 位寿司, 2 位非食事)



(g) カテゴリー外のものを撮影。



(h) 複数品目 (味噌汁 1 位, 牛丼 5 位)

図 3 DeepFoodCam 実行例。

カロリー推定を行うことも予定している。なお、Android のアプリの APK ファイルは、FoodCam の日本語公式サイト*5 で公開する予定である。

参考文献

- [1] Jin, J., Dunder, A. and Culurciello, E.: Flattened Convolutional Neural Networks for Feedforward Acceleration, *Proc. of International Conference on Learning Representation Workshop Track* (2015).
- [2] Kawano, Y. and Yanai, K.: FoodCam: A real-time food recognition system on a smartphone, *Multimedia Tools and Applications*, pp. 1–25 (2014).
- [3] Kawano, Y. and Yanai, K.: ILSVRC on a Smartphone, *IPSJ Transactions on Computer Vision and Applications*, Vol. 6, pp. 83–87 (2014).
- [4] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks., *Advances in Neural Information Processing Systems* (2012).
- [5] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I. and Lempitsky, V.: Speeding-up Convolutional Neural Networks Using Fine-tuned CP-Decomposition, *Proc. of International Conference on Learning Representation Conference Track* (2015).
- [6] Lin, M., Chen, Q. and Yan, S.: Network In Network, *Proc. of International Conference on Learning Representation Conference Track* (2014).

*5 <http://foodmobi.jp/>