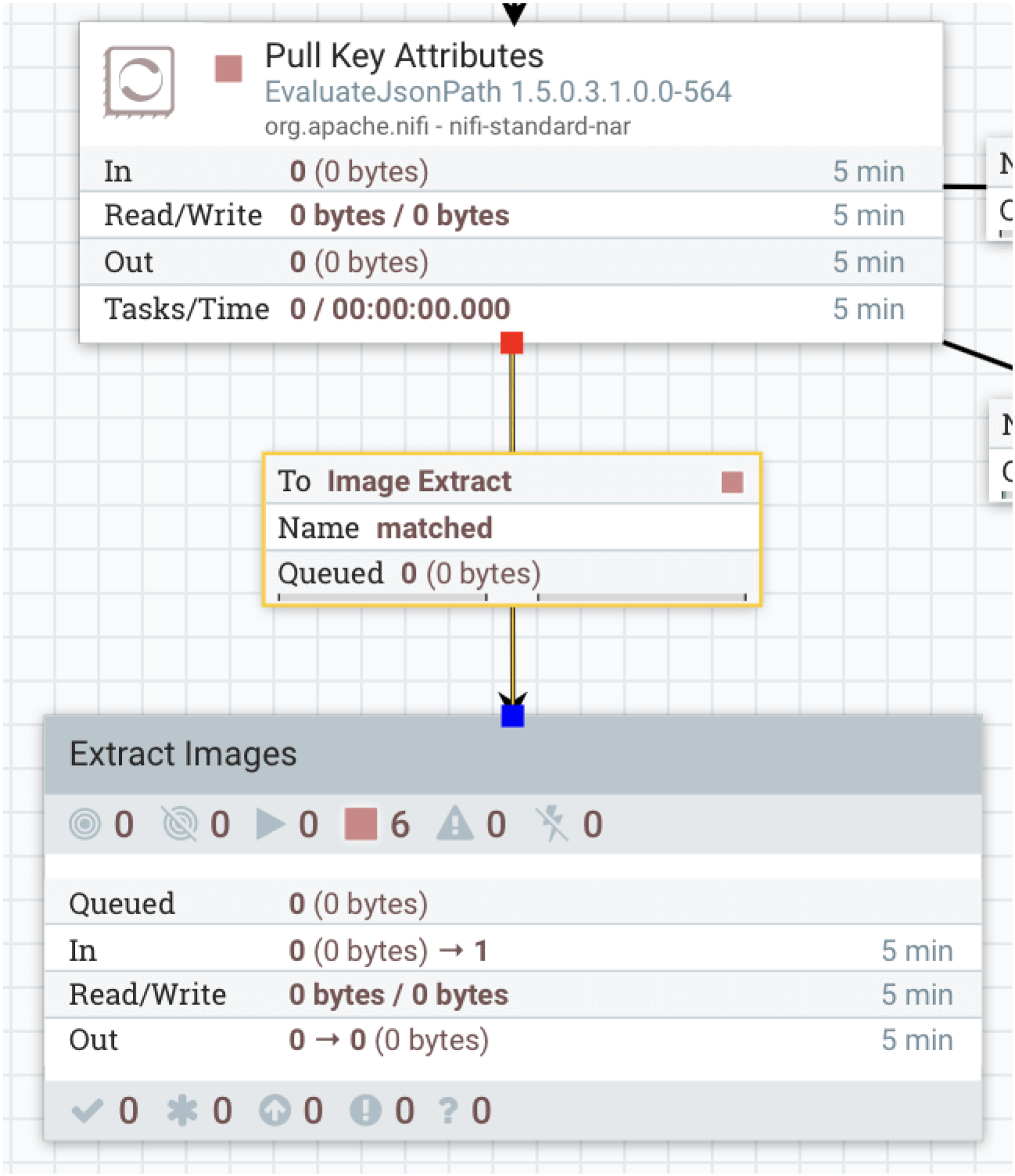
FlashBlade でシンプルに! ビッグデータ& AI –
深層学習モデルのトレーニングと展開



愛犬家のためのAIの構築を例に、FlashBlade で構築するエンドツーエンドの AI パイプラインをご紹介するブログシリーズ。第 1 回ではアーキテクチャの概要を、第 2 回ではリアルタイムなデータの取得、処理、視覚化がいかに容易にできるかを、続く第 3 回では FlashBlade と Apache Spark、Zeppelin、Apache Hive を使用したビッグデータ分析についてご説明しました。最終回となる今回は、深層学習モデルのトレーニングと展開についてご説明し、そのモデルを使用して実際に犬画像を検出してみます。

深層学習モデルのトレーニング

愛犬家のための AI の中心にあるのは、犬の画像を検出するための YOLO（You Only Look Once）と呼ばれる深層学習モデルです。YOLO は精度が高く、リアルタイムで実行できるため、よく利用されています。

一般的に、深層学習システムの構築は次のような手順で進めます。

1. 大量のデータを収集する
2. 学習用データセットにラベル付けをする
3. モデルのトレーニング、検証、調整を行う
4. モデルを展開する
5. モデルのパフォーマンスを測定し、フィードバックを適用し、モデルを更新する

ツイートから犬の画像を抽出する

深層学習モデルのトレーニングとテストには、膨大な量のデータが必要です。ここでは Apache NiFi を使用して Twitter API から犬の画像をダウンロードします。

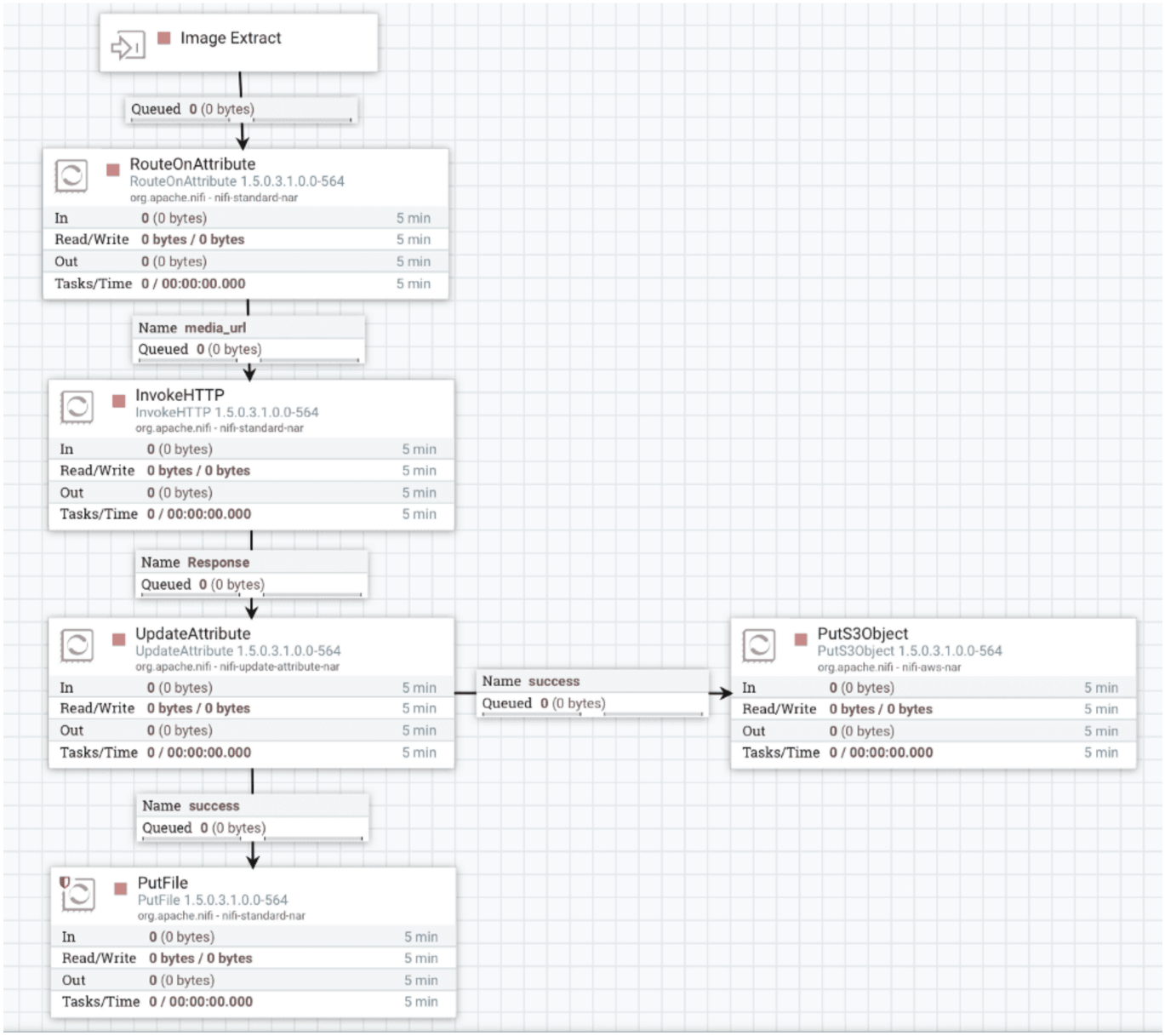


図 1: NiFi を使用してツイートから犬の画像を抽出

このサブフローはツイートを評価します。ツイートに画像がある場合、NiFi は HTTP リクエストを呼び出して画像をダウンロードし、FlashBlade NFS マウントと S3 バケット（デモ用）に保存します。

NiFi UI 上で Extract Images サブフローをメインフローに接続します。下図に示すように、私はこれを Pull Key Attributes プロセッサから Extract Images サブフローにドラッグし、ポップアップされた「matched」を選択することで実行しました。

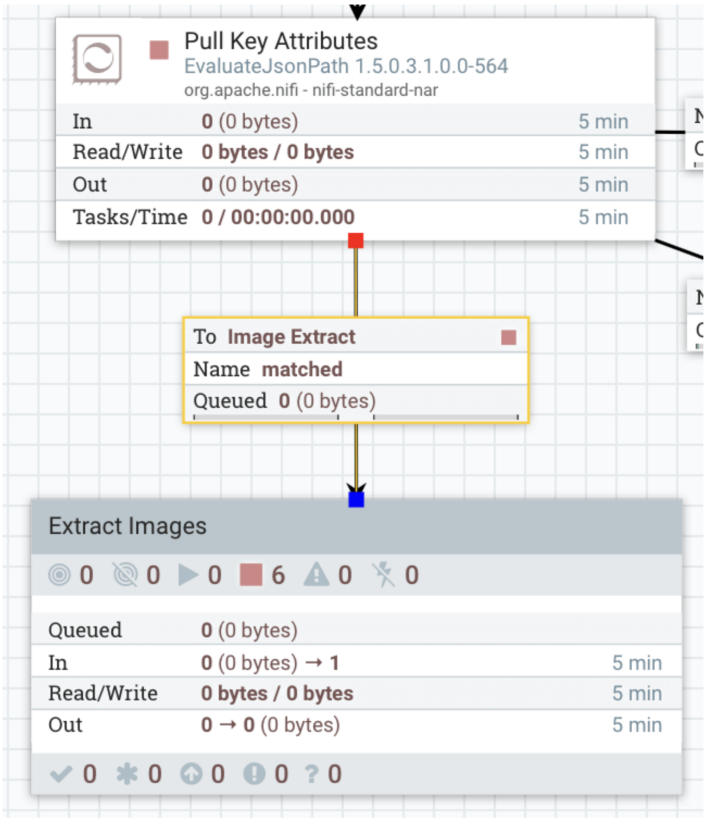


図 2: Extract Images サブフローをメインフローに接続

フローを接続すると、NiFi は画像の抽出を開始します。FlashBlade NFS ボリュームと S3 バケットに画像が保存されることを確認できます。ダウンロードした犬の画像を見るために、Apache Ambari から S3 Viewer を開きます。

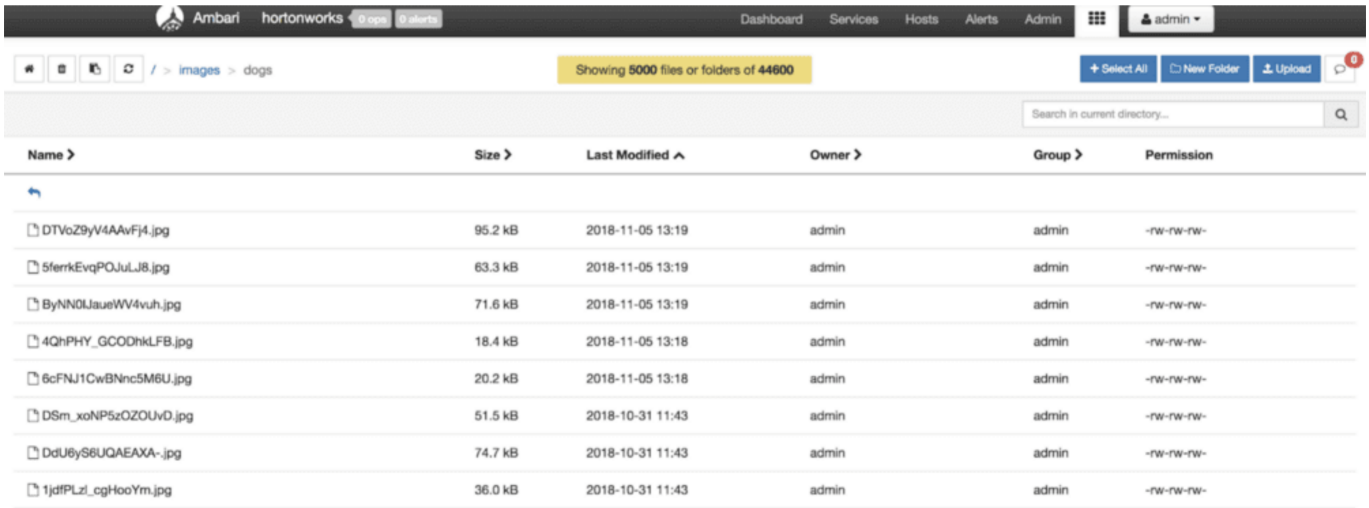
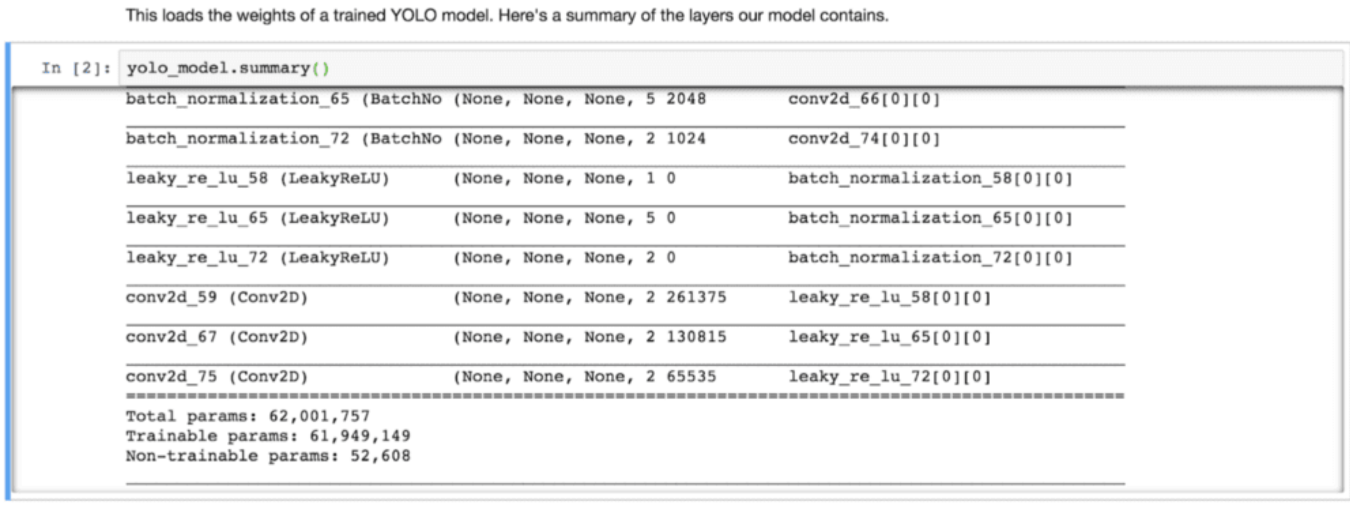


図 3: Ambari の S3 Viewer



深層学習モデルをトレーニングする

データを取得したら、データにラベルを付け、ラベル付きデータを使用して深層学習モデルのトレーニングを行います。YOLO モデルのトレーニングでは、ラベルの付いた膨大な量のデータから学習することによって、犬がどのような形をしているものかを理解しようとします。また、モデルがどの程度正確にオブジェクトを検出しているかを検証し、パフォーマンスを調整します。このプロセスは通常、繰り返し行われます。深層学習モデルのラベル付け、トレーニング、検証方法についての詳しい説明は、本ブログの趣旨とは異なるため、ここでは割愛します。

YOLO モデルのトレーニングには非常に長い時間を要し、広範囲のターゲットオブジェクトクラスを得るために、ラベル付き境界ボックスのデータセットを大量に必要とします。ここでは、Jupyter のノートブックで、事前にトレーニングされた YOLO モデルをロードし、そのサマリーを表示してみます。
[crayon-6424c64a9f96e889660740/]

This loads the weights of a trained YOLO model. Here's a summary of the layers our model contains.

```
In [2]: yolo_model.summary()
batch_normalization_65 (BatchNo (None, None, None, 5 2048        conv2d_66[0][0]
batch_normalization_72 (BatchNo (None, None, None, 2 1024        conv2d_74[0][0]
leaky_re_lu_58 (LeakyReLU)      (None, None, None, 1 0           batch_normalization_58[0][0]
leaky_re_lu_65 (LeakyReLU)      (None, None, None, 5 0           batch_normalization_65[0][0]
leaky_re_lu_72 (LeakyReLU)      (None, None, None, 2 0           batch_normalization_72[0][0]
conv2d_59 (Conv2D)              (None, None, None, 2 261375      leaky_re_lu_58[0][0]
conv2d_67 (Conv2D)              (None, None, None, 2 130815      leaky_re_lu_65[0][0]
conv2d_75 (Conv2D)              (None, None, None, 2 65535       leaky_re_lu_72[0][0]
==================================================================================================
Total params: 62,001,757
Trainable params: 61,949,149
Non-trainable params: 52,608
```

図 4: YOLO モデルのサマリー

このモデルには、6,200万以上のパラメータがあります。そのため、学習を高速化してデータサイエンティストの生産性向上を図るには、超並列コンピューティングが可能な GPU と超並列 I/O が可能な FlashBlade が必要なのです。

深層学習モデルの展開

モデルのパフォーマンスに満足できたという想定で、いよいよこのモデルを展開し、犬の画像の検出を行います。本番環境では通常、モデルを REST API として展開します。そうすることで、アプリケーションが簡単に API にアクセスしてモデルを統合し、オブジェクトを検出できるようになります。

REST API としてモデルを展開する

Python サーバーを起動し、YOLO モデルをロードして HTTP を介して提供します。

[crayon-6424c64a9f978019509656/]
API サーバーは、localhost:8080 をリッスン（待機）します。これで画像から犬を検出する準備が整いました。API サーバーに画像を送信するためのスクリプトを実行します。
[crayon-6424c64a9f97b433536522/]
API の応答から、画像に犬がいることと、どこに写っているのかがわかります（境界ボックスで示されます）。このモデルの検出率は 98% です。モデルが検出した右側の画像では、犬を囲むように緑色のボックスが表示されています（図5）。

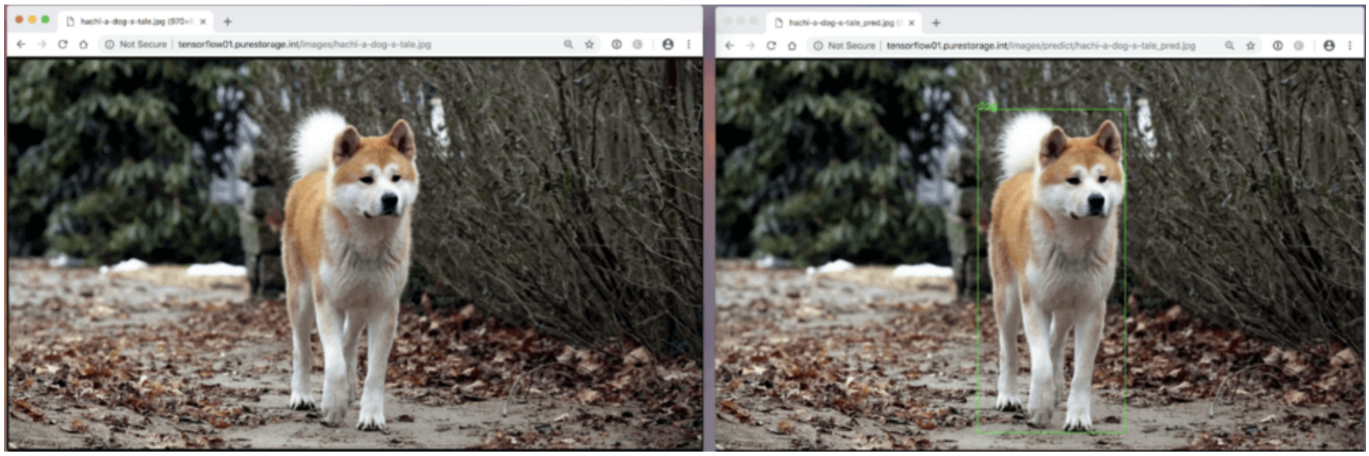


図 5: オリジナルの画像と検出された画像
（出典: ソニー・ピクチャーズ・エンタテインメント『HACHI 約束の犬』より）

モデルをエッジデバイスに展開する

Raspberry Pi などのエッジデバイスにモデルを展開することも可能ですが、通常ではエッジデバイスには十分なコンピューティング能力がないため、Intel Neural Compute Stick 2 や Google Edge Tensor Processing Unitを使用して高速化を図ります。これにより、ライブ動画から犬の画像をリアルタイムに検出することができるようになります。このトピックについては、別のブログを書こうと思っています。

まとめ

このブログシリーズを通して、FlashBlade を利用することで、サイロに完全になくすことが可能となり、複雑な管理作業から開放されるということをお伝えしてきました。FlashBlade の登場以前は、AI パイプラインの各段階でサイロ化されたデータシステムを使用する必要がありました。データレイク用には HDD の DAS（直接接続ストレージ）、ストリーミング分析用には別の DAS（HDD/SSD）、深層学習や GPU システム用には HPC のストレージといった具合です。

FlashBlade を使用すれば、これら全てのアプリケーションを単一のプラットフォームに統合できます。統合されたデータはアプリケーション間で共有され、より高度な洞察の抽出を可能にします。FlashBladeではシンプルな使い勝手と優れた柔軟性を実現しており、アプリケーションのニーズに応じて自在にリソースを利用することができます。

FlashBlade はお客様の環境に必ず変革をもたらします。

ハッピーコーディング!

シリーズ「**FlashBlade** でシンプルに! ビッグデータ & **AI**」

FlashBlade でシンプルに! ビッグデータ & AI：第 1 回

FlashBlade でシンプルに! ビッグデータ & AI：第 2 回 — リアルタイムなデータ取得とデータ処理

FlashBlade でシンプルに! ビッグデータ & AI：第 3 回 — ビッグデータの分析