バイオ医薬品および医薬品開発の分野におけるインテルのエッジ AI テクノロジー

この記事は、Medium に公開されている「Intel Edge AI Technology in the Realm of Biopharma and Drug Development」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

バイオ医薬品向け AI パイプラインの作成: 洞察と課題

進化し続けるバイオ医薬品技術と医薬品開発の状況において、**モノクローナル抗体産生**の**細胞分析**分野における最近の取り組みにより、ソリューションの拡張と生産における複雑な課題を克服する上での**エッジ AI テクノロジー**の重要な役割が明らかになりました。

インテルは、さまざまなパートナーと協力してこのプロセスに関与してきました。**細胞画像プロジェクト**に対するインテルの貢献の 1 つは、複数のディープラーニング・モデルを含む AI パイプラインを使用した明視野像の処理を中心に展開されています。パイプラインの目的は、細胞およびその他の生物学的要素を特定し、細胞の形態、生存率、表現型の変化などの動的な生物学的特性に関するフィードバックを提供することです。

cell-AI プロジェクトに取り組むと、通常、一連の固有の課題がまず浮かびます。

最初に、これは学際的な分野であり、データ・サイエンティストとバイオ医薬品のエキスパートの間には知識のギャップがあるため、計画と妥当性のチェックのために、より明確なコミュニケーションが必要になります。研究室で AI ソリューションを実装しようとするときに、データ・サイエンティストとベンチ・サイエンティスト (研究室で働く科学者) が、互いの役割の性質とニーズを把握するのに苦労することがよくあります。この相互理解の欠如が、多様な研究室の環境に統合する必要がある AI ソリューションの使いやすさと拡張性を妨げる可能性もあります。

2 つ目の課題は、機器のばらつきです。プレートリーダー[1] 顕微鏡が異なると、ハードウェア、光学系、絞りが異なるため、生成される画像に一貫性がなくなり、途中でこれらの不一致を評価して対処するための追加の作業が必要になります (定期的な追跡されたキャリブレーションや調整など)。さらに、機器のベンダー間の違い、培養温度、培地条件、遺伝子組み換えはすべて、データの変動性やディープラーニング・パイプラインの固有の転送可能性に影響を与える可能性があります。その結果、エッジおよびクラウドの MLOps コンポーネントで DL モデルのパフォーマンスを監視する必要性が生じます。

3 つ目の課題は、査読ラベルの取得です。このプロセスは教師ありマシンラーニングに基づいており、クリーンで正確なラベルの取得には非常にコストと時間がかかります。

最後の課題はモデルのデプロイです。ほとんどの場合、データサイズとデータ・プライバシーの理由から、クラウドでのデプロイはオプションになりません、プレートリーダー[1] 顕微鏡から生成される画像は巨大であり、データをクラウドに転送して結果を戻すと、大量のデータのストリーミングが必要になるため (1 時間あたり 30GB)、レイテンシーが大きくなります。そしてさらに重要なことは、研究室は通常、データの共有に消極的なことです。これらの 2 つの制約のため、通常、クラウドでのデプロイはオプションにならず、パイプラインをエッジにデプロイする必要があります。

上記の課題を考慮して、細胞治療および初期段階の医薬品開発分野における一般的なユースケースと、提案された AI パイプラインについて話すことにしましょう。

CHO 細胞セグメンテーションのユースケース

CHO 細胞はチャイニーズ・ハムスター卵巣細胞の略で、モノクローナル抗体 (MABS)、融合タンパク質、ホルモン、凝固因子などを含む、非常に大きく複雑なタンパク質分子を生成する能力があり、タンパク質合成に適しています。そのため、CHO 細胞では、細胞が「生成物」である幹細胞や CAR-T 細胞とは異なり、生成するタンパク質が「生成物」です。

商業的タンパク質生産の一環として、**生存率**と**生産能力**を測定する必要があります。研究室では、測定は、グルコース濃度、温度、溶存酸素、pH などの評価を使用して間接的に行われます。細胞を直接測定するには、培養と染色が必要です。

以下にワークフローを示します。

1. 培養 — 細胞を培養します。
2. 細胞の固定 — 高価な試薬で洗浄して培地を除去します。
3. 透過処理 - より高価な化学物質で洗浄して細胞膜を透過処理します (細胞間タンパク質を染色するため)。
4. ブロッキング — 細胞を別の高価な試薬中でインキュベートし、特定の抗体が結合しないようにします。
5. 一次抗体のインキュベーション — 抗体を生成されるタンパク質に特異的に結合します。
6. 洗浄 - より高価な化学物質を使用して結合していない一次抗体を除去します。
7. 核染色 — DAPI などの核染色を使用して細胞核を視覚化し、洗浄ステップと同じ化学物質で洗浄します。
8. マウント — 顕微鏡 (プレートリーダー[1]) で読み取る準備をします。
9. イメージング – 染色された細胞をカウントし、タンパク質生成サイクルの状態と相対的な細胞の健康状態を判断します。最終的にはタンパク質の生成が途絶えて生成が停止し、バッチをフラッシュする必要があります (細胞数、生存数などは画像ではなく出力です)。

複数のディープラーニング・モデルとデータの前処理/後処理を含む AI パイプラインを使用すると、ステップ 1 から直接ステップ 9 に進むことができ、染色ワークフローから実用的な結果を得る際の労力とレイテンシーの大部分を排除し、高価な特殊化学物質の要件を回避できます。インテルは、細胞画像プロジェクト (https://www.cellimage.ie/ (英語)) の一部として、上記のパイプラインをデプロイし、エッジ上でこれらの画像を推論するリファレンス実装をまとめました。この設計には、**OpenVINO™ ツールキット**、**OpenVINO™ モデルサーバー**、**AI Connect for Scientific Data (AiCSD)** を使用しました。これらの優れたソフトウェア・パッケージについてそれぞれ簡単に説明します。

細胞治療ソリューションに適用されるインテルのソフトウェア・パッケージ

インテル® ディストリビューションの OpenVINO™ ツールキットは、汎用インテル® アーキテクチャー上で包括的なディープラーニング推論を最適化、チューニング、実行します。エッジからクラウドまで、高いコンピューティング・パフォーマンスと豊富なデプロイオプションを提供し、この数年で、複数のツール、リポジトリー、コンポーネントを備えた非常に広範なエコシステムに成長しました。

OpenVINO™ ツールキットの中心には、モデルをロードして実行する **OpenVINO™ ランタイム**があります。ランタイムは、プラグインを使用して、ディープラーニング・モデルがインテルのハードウェア上で行う低レベルの操作を効率良く実行します。CPU プラグイン、GPU プラグイン、ヘテロジニアス・プラグインなど、ハードウェアごとに異なるプラグインを用意しています。

CPU プラグインは、ディープ・ニューラル・ネットワーク向けインテル® マス・カーネル・ライブラリー (インテル® MKL-DNN) を使用して、CPU 上でハイパフォーマンスなニューラル・ネットワークを実現します。

GPU プラグインは、ディープ・ニューラル・ネットワーク向けコンピュート・ライブラリー (clDNN) を使用して、GPU 上でディープ・ニューラル・ネットワークを推論します。

ヘテロジニアス・プラグインを使用すると、複数のデバイス上で 1 つのネットワークの推論を計算できます。ヘテロジニアス・モードでネットワークを実行する目的は次のとおりです。

- アクセラレーターの能力を利用してネットワークの最も重い部分を処理し、CPU などのフォールバック・デバイス上でサポートされていないレイヤーを実行する。
- 1 回の推論中に、利用可能なすべてのハードウェアを効率良く利用する。

OpenVINO™ ツールキットのもう 1 つの部分は、モデルを最適化し、TensorFlow*、PyTorch*、ONNX* などの一般的なディープラーニング・フレームワークから **OpenVINO™ 中間表現形式**に変換する**モデル・オプティマイザー**です。モデルは、量子化、固定、融合などの手法を使用して最適化されます。モデルは、オンプレミスとオンデバイス、ブラウザー、クラウドの、インテルのハードウェアと環境の組み合わせにデプロイできます。

OpenVINO™ は、推論に加えて、トレーニング中にモデルに圧縮アルゴリズムを実装するニューラル・ネットワーク圧縮フレームワーク (NNCF) (英語) ツールを提供します。

図 1: OpenVINO™ の概要。OpenVINO™ の詳細なドキュメントは、openvinio.ai (英語) を参照。

OpenVINO™ モデルサーバー (OVMS)

デプロイには、OpenVINO™ ランタイムを使用することも、OpenVINO™ モデルサーバー (OVMS) を使用することもできます。

OVMS は、AI モデルとパイプラインを提供するための、スケーラブルでハイパフォーマンスなツールです。AI モデルの管理を一元化できるため、多数のデバイス、クラウド、コンピューティング・ノード間で一貫した AI モデルを維持するのにも役立ちます。簡単に言えば、OVMS とはモデルを**ロード**し、**管理**し、**ネットワーク API を通じて**その機能を公開する**マイクロサービス**であり、システム内のほかのコンポーネントは、それらを操作してモデルを利用できます。OVMS は、TensorFlow* Serving と KServe 互換の 2 種類の API を公開しており、どちらも gRPC または RESTful API2 を介して推論、モデルステータス、モデル・メタデータ・サービスを提供します。

OpenVINO™ ランタイムの代わりに OVMS を使用するのはどういった場合でしょうか? まず、ほかのオプションがない場合です。OpenVINO™ は C++ プロジェクトであり、公式の Python* バインディングがありますが、ソフトウェア・スタックが別の言語の場合は OVMS を使用します。次に、インターフェイスの実装が困難な場合です。OVMS には OpenVINO™ をシステムに含めるために必要な機能が用意されているため、作業を簡素化できます。また、システム・ソリューションがすでにマイクロサービス・パラダイムで動作している場合も、OVMS を使用する選択が適切なことは明白です。ほかのコンポーネントのビジネスロジックに OpenVINO™ を含めてアプリケーションを拡張したくない場合や、ビルドシステムに手を加えたくない場合もあります。アプリケーションの一部がモバイルなどのあまり強力ではないデバイスで実行されていて、それらのデバイスに大きな推論負荷をかけたくない場合は、より強力なマシンに計算を渡す必要があります。OVMS でネットワーク API を公開すると、コンポーネントを複数のデバイス上で実行でき、リクエスト形式でデータを OVMS に送ると、レスポンスとしてモデル出力が返されます。

ネットワーク API と OVMS はマイクロサービスであるという事実により、**ソリューションのスケールアップ**に適しています。例えば、マルチノードの Kubernetes* クラスターがある場合、複数のレプリカを作成し、それらの前にロードバランサーを設定して、単一ノードの能力を超える高可用性と高いスループットを実現できます。このようなシステム全体の集約は、OVMS を使用すると簡単に実現できます。

これらに加えて、**セキュリティーとプライバシー目的**で OVMS を使用すると、信頼できるマシン上でモデルサーバーをホストできるようになり、内部または外部からサーバーにアクセスするほかのすべてのアプリケーションはモデルを認識できなくなります。ほかのアプリケーションにインターフェイスを公開するだけで、ほかのアプリケーションはモデルにアクセスしたり、モデルを表示することはできません。

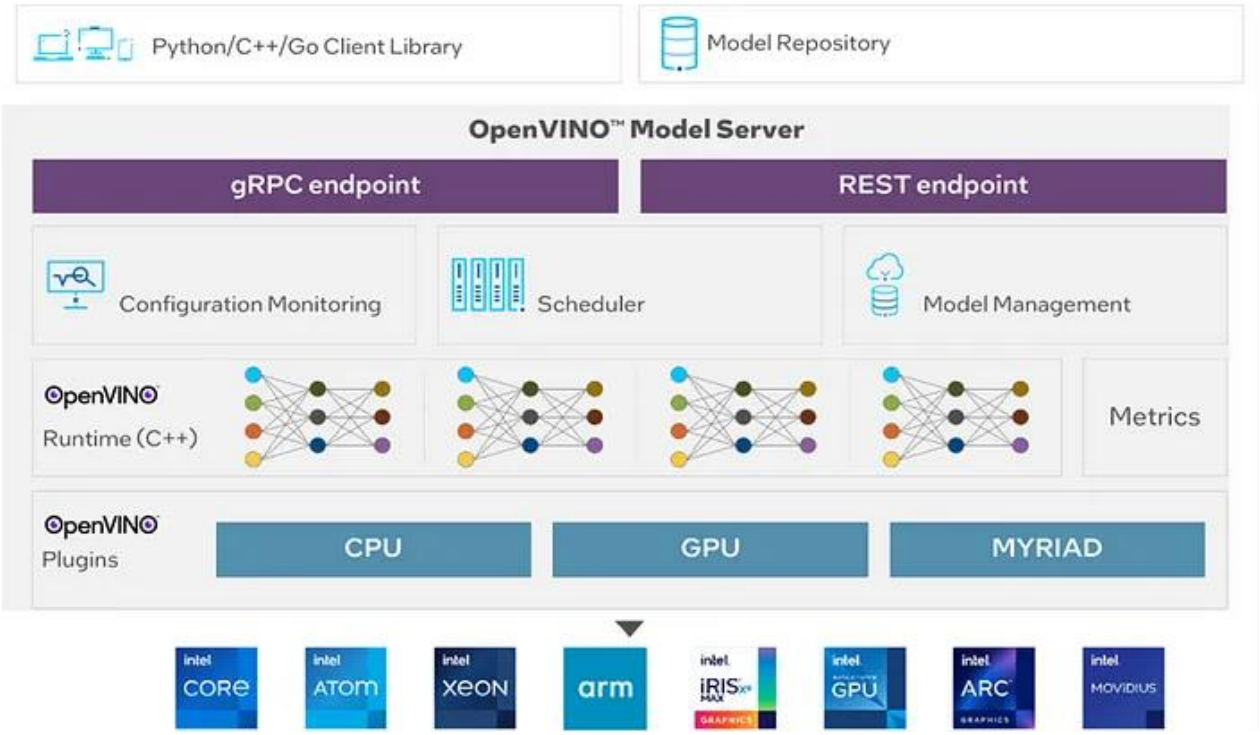

図 2: OpenVINO™ モデルサーバー

OVMS の構造を見てみましょう (図 2)。上部から、ネットワーク・インターフェイスがあり、gRPC と Restful エンドポイントがあります。両方のエンドポイントで TensorFlow* Serving API と KServe API をサポートしており、これらの API を使用して推論を呼び出すことができます。メタデータを呼び出して、モデルがどのような種類の入力を予測しているか、モデルからどのような出力が予測されるかについての情報を得ることもできます。

もう 1 つの特徴は、複数のモデルを同時に提供できることです。構成ファイルでこれらを指定すると、OVMS が**モデル管理**を行います。モデルサーバーはモデルファイルの場所を監視し、モデルのバージョン管理もサポートします。特筆すべきもう 1 つの点は、モデルの場所がローカル・ファイル・システムである必要がないことです。OVMS は、Google Cloud、AWS S3、Azure などのリモート・**ストレージ・ファイル・システム**をサポートします。

AI Connect for Scientific Data (AiCSD) (英語)

AI Connect for Scientific Data (AiCSD) は、科学機器からのデータを AI パイプラインに接続し、**エッジ**でワークロードを実行するオープンソース・ソフトウェア・サンプルです。

画像処理と**自動画像比較**のパイプラインも管理します。AiCSD は、オープンソースの **EdgeX サービス**を利用してコンテナ化されたマイクロサービス・ベースのソリューションであり、セキュアな **Redis メッセージブローカー**とさまざまな通信 API により接続されていて、さまざまなユースケースや設定に適応できます。図 3 は、このリファレンス実装向けに作成されたサービスを示しています。

AiCSD のアーキテクチャー・コンポーネントには次のものが含まれます。

- **マイクロサービス**: インテルにより提供されるマイクロサービスには、ファイルとジョブを管理するためのユーザー・インターフェイスとアプリケーションが含まれます。
- **EdgeX アプリケーション・サービス**: AiCSD は、EdgeX アプリケーション・サービスの API を使用して情報の通信と転送を行います。
- **EdgeX サービス** (英語): サービスには、データベース、メッセージブローカー、セキュリティー・サービスが含まれます。
- **パイプラインの実行**: AiCSD は、パイプライン管理用のサンプル・パイプラインを提供します。
- **ファイルシステム**: AiCSD は、入力ファイルと出力ファイルを保存および管理します。
- **サードパーティーの入力デバイス**: デバイスは、処理される画像を提供します。例えば、光学顕微鏡やベルト・コンベアー・カメラなどです。

リファレンス・アーキテクチャーでは、割り当てられたジョブを使用して画像を処理できます。ジョブは、ファイルの移動、ステータス、パイプラインからの結果や出力を追跡します。一部のタスクは、ジョブに関する情報と実行する適切なパイプラインを照合するのに役立ちます。

このプロセスは次のように説明できます。

1. **入力デバイス/イメージャー**は、ファイルを**ファイル・ウォッチャー**により監視されているディレクトリーの **OEM ファイルシステム**に書き込みます。ファイル・ウォッチャーは、ファイルを検出すると、HTTP リクエスト経由でジョブ (特定フィールドの JSON 構造体) を**データ・オーガナイザー**に送ります。
2. **データ・オーガナイザー**は、ジョブを**ジョブ・リポジトリー**に送り、新しいジョブを **Redis データベース**に作成します。ジョブ情報が**タスクランチャー**に送られ、ジョブと一致するタスクがあるかどうか判断されます。一致するタスクがある場合、ジョブは**ファイルセンダー (OEM)** に進みます。
3. **ファイルセンダー (OEM)** は、ジョブとファイルの両方を**ファイルレシーバー (ゲートウェイ)** に送る責任を負います。**ファイルレシーバー (ゲートウェイ)** がファイルをゲートウェイ・ファイル・システムに書き込むと、ジョブは**タスクランチャー**に送られます。
4. **タスクランチャー**は、**EdgeX メッセージバス (Redis 経由)** を使用してジョブを適切なパイプラインに送る前に、ジョブと一致するタスクがあることを確認します。ML パイプラインは適切なトピックをサブスクライブして、そのパイプラインでファイルを処理します。出力ファイル (存在する場合) はファイルシステムに書き込まれ、ジョブは**タスクランチャー**に送り返されます。
5. 次に、タスクランチャーは、出力ファイルがあるか、結果のみがあるかを判断します。結果のみで出力ファイルがない場合、タスクランチャーはジョブを完了としてマークします。出力ファイルがある場合、**タスクランチャー**はジョブを**ファイルセンダー (ゲートウェイ)** に送ります。
6. **ファイルセンダー (ゲートウェイ)** は、**ファイルレシーバー (OEM)** がサブスクライブおよびプルできるように、Redis 経由でジョブ情報を **EdgeX メッセージバス**に公開します。ファイルレシーバー (OEM) は、出力ファイルの HTTP リクエストを**ファイルセンダー (ゲートウェイ)** に送ります。ファイルが応答の一部として送られ、**ファイルレシーバー (OEM)** は出力ファイルをファイルシステムに書き込みます。

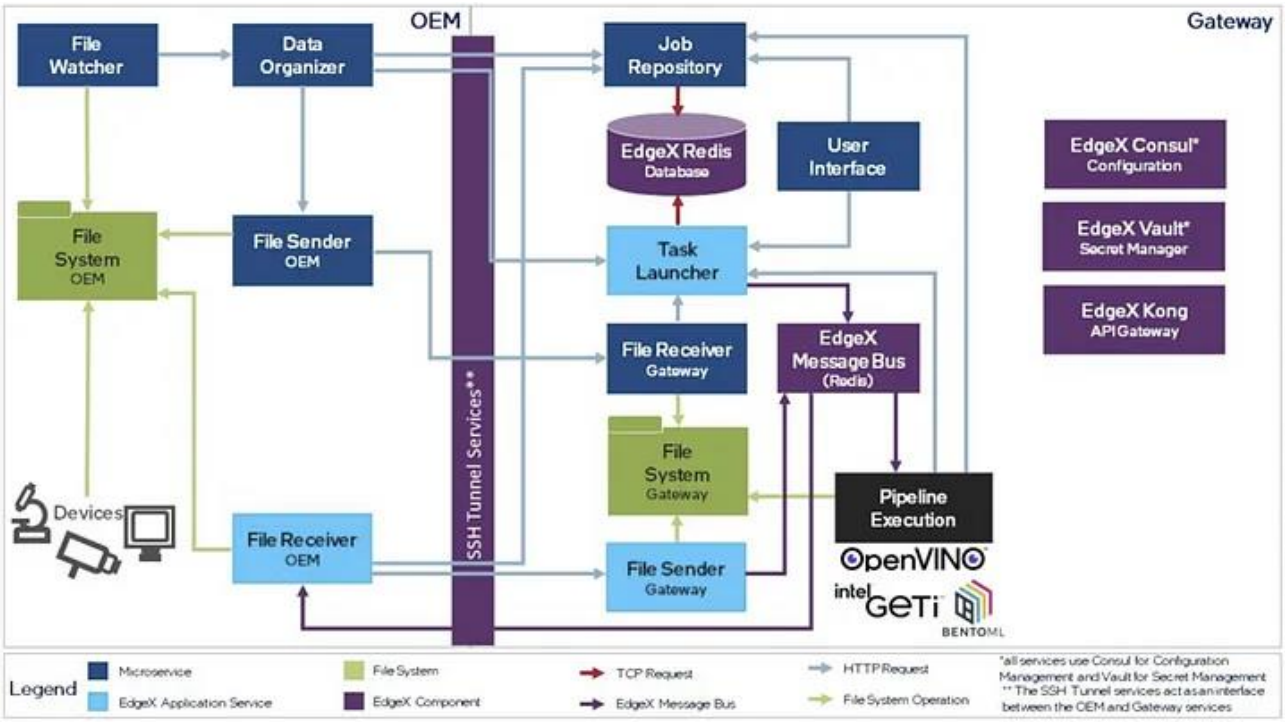

図 3: アーキテクチャーと高レベルのデータフロー

CHO 細胞セグメンテーション向け AI パイプラインのユースケース

顕微鏡 (プレートリーダー[1]) はローカル・ファイル・システムにデータファイルを生成するため、分析するには AI ソフトウェアおよびハードウェア・リソースが利用可能な別の機器にデータファイルを転送する必要があります。このデータとモデルの場所の相違に対処するには、柔軟なマイクロサービス・ベースのソリューションが必要です。AiCSD は **EdgeX Foundry マイクロサービス**を利用して科学機器データの自動検出、管理、処理を行っているため、AiCSD エッジ・マイクロサービス・インフラストラクチャーを使用してデータを**エッジ・コンパニオン・コンピューティング・デバイス**に移動します。このマイクロサービスの柔軟性は、このプロジェクトに固有のヘテロジニアス・システム統合と非対称データ・インターフェイスに対処する上で重要です。

複雑な AI パイプラインには、画像の前処理ステップ、OpenVINO™ ツールキットにより最適化された複数のディープラーニング・モデルの推論、画像の後処理ステップが含まれており、これらはすべて別のオープンソース・ツール **BentoML** (英語) を使用してコンテナ化されています。OpenVINO™ ツールキットを使用すると、推論のレイテンシーが小さくなり、プロセスが高速化されます。最終結果 (細胞のセグメンテーションと生存率の検出結果) は、エッジ・コンパニオン・コンピューティング・ファイル・システムで生成されます。次に、**EdgeX メッセージバス** (AiCSD マイクロサービス・インフラストラクチャーの一部) に、結果を元の科学機器のローカル・ファイル・システムにコピーし、必要に応じて保存するように通知します。

図 4 は、DL モデルを使用して細胞画像をプロセスする例を示しています。この例では、UNet を使用して MSC 核をマスクおよびカウントしています。

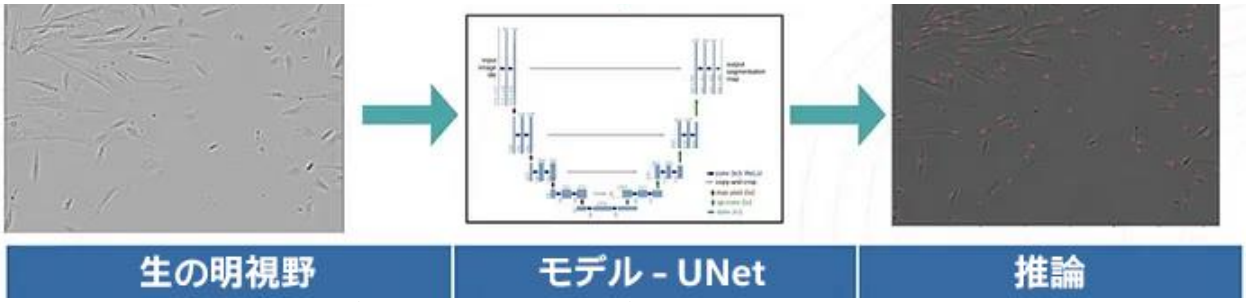

図 4: UNet ディープラーニング・モデルを使用した MSC 核のカウント。

参考資料:

1. プレートリーダーは、マイクロタイター・プレート内のサンプルから画像を取得するために使用される実験器具です。リーダーは、プレートのウェル内のサンプルを通して、特定のキャリブレーションされた周波数の光 (UV、可視光、蛍光など) を照射します。プレートリーダー顕微鏡データセットには固有の変動性があり、定期的な追跡されたキャリブレーションと調整が必要です。
2. https://docs.openvino.ai/archive/2023.2/ovms_what_is_openvino_model_server.html (英語)

OpenVINO™ ツールキットとは

AI を加速する無償のツールである OpenVINO™ ツールキットは、インテルが無償で提供しているインテル製の CPU や GPU、VPU、FPGA などのパフォーマンスを最大限に活用して、コンピューター・ビジョン、画像関係をはじめ、自然言語処理や音声処理など、幅広いディープラーニング・モデルで推論を最適化し高速化する推論エンジン/ツールスイートです。

OpenVINO™ ツールキット・ページでは、ツールの概要、利用方法、導入事例、トレーニング、ツール・ダウンロードまでさまざまな情報を提供しています。ぜひ特設サイトにアクセスしてみてください。

https://www.intel.co.jp/content/www/jp/ja/internet-of-things/openvino-toolkit.html

法務上の注意書き

インテルのテクノロジーを使用するには、対応したハードウェア、ソフトウェア、またはサービスの有効化が必要となる場合があります。

絶対的なセキュリティーを提供できる製品またはコンポーネントはありません。

実際の費用と結果は異なる場合があります。