

# 畳み込みニューラルネットワークによる画像認識技術のマウンテンプロット 画像への適用

## APPLYING IMAGE RECOGNITION TECHNOLOGY BY CONVOLUTIONAL NEURAL NETWORKS TO MOUNTAIN PLOT IMAGES

野村昌弘 \*<sup>A)</sup>、田村文彦 <sup>A)</sup>、島田太平 \*<sup>A)</sup>、山本昌亘 <sup>A)</sup>、  
古澤将司 <sup>B)</sup>、杉山泰之 <sup>B)</sup>、原圭吾 <sup>B)</sup>、長谷川豪志 <sup>B)</sup>、大森千広 <sup>B)</sup>、吉井正人 <sup>B)</sup>  
Masahiro Nomura\*<sup>A)</sup>, Fumihiko Tamura<sup>A)</sup>, Taihei Shimada<sup>A)</sup>, Masanobu Yamamoto<sup>A)</sup>,  
Furusawa Masashi<sup>B)</sup>, Yasuyuki Sugiyama<sup>B)</sup>, Keigo Hara<sup>B)</sup>, Katsushi Hasegawa<sup>B)</sup>,  
Chihiro Ohmori<sup>B)</sup>, Masahito Yoshii <sup>B)</sup>

<sup>A)</sup>Japan Atomic Energy Agency, JAEA

<sup>B)</sup>High Energy Accelerator Research Organization, KEK

### Abstract

Image recognition using a convolutional neural network (CNN) has been used in a wide range of fields and has produced excellent results. If this image recognition technology is used effectively, it should be possible to obtain information from an image equal to or more than the information that a person can obtain from an image. At the J-PARC, researchers with specialized knowledge obtain beam information needed to adjust the equipment from an image called mountain plot. In this study, we applied the image recognition technology by using CNN to this mountain plot image, and tried to obtain the information about the beam necessary for adjustment. As a result, we were able to obtain more information than is currently available by using the image recognition technology. In the future, we plan to adjust the equipment based on the information actually obtained from the image recognition technology and confirm its effectiveness.

### 1. 序

畳み込みニューラルネットワーク (Convolutional Neural Network:CNN) による画像認識の技術は、幅広い分野で用いられ、優れた結果を残している。この画像認識の技術を上手く利用すれば、人が画像から得られる情報と同等かそれ以上の情報を、CNNにより画像から情報を得ることができるはずである。画像から情報を得ることに関しては、J-PARCではマウンテンプロットと呼ばれる画像から、専門知識を持った人が情報を得て、加速器立上げ時等に機器の調整を行なっている。マウンテンプロットとは、RCS内に設置された Wall Current Monitorの波形を周回毎にスライスし、縦に並べて表示した画像であり、様々なビームの情報、前段の Linac から RCS への入射運動量や入射タイミング、入射ビームの運動量広がり等の情報を含んでいる。

本研究では、CNNによる画像認識の技術をこのマウンテンプロットに適用し、RCSの調整時に用いている、RCSへの入射タイミングと入射運動量について、更に、マウンテンプロットからでは難しい入射運動量広がりについても求めてみた。具体的に行った内容としては、まず、シミュレーションによりビーム情報を教師データとした学習用画像を作成し、次に、CNNに学習させることにより、画像から教師データが求められる様にした。そしてこれらを行なった後、実際のマウンテンプロットに適用し、学習済みCNNにより先に述べた情報を求めた。

### 2. ニューラルネットワーク

今回画像認識に用いたCNNは、参考文献 [1] のサンプルソースコードをほぼそのまま採用した21層からなるネットワークである。ネットワークの構成を Figure 1 に示す。入力サイズは400×120、縦と横をそれぞれ1/2にするPooling層が3層あるため、最終のPooling層の出力サイズは1/8の50×15となる。Filter sizeは3、Paddingの大きさは1と2、Strideの大きさは1に設定している。これらのパラメータに関しては、サンプルソースコードそのままの値を使用している。Figure 1 に示すように Over fittingを防ぐために dropout の層を挿入しているが、学習用画像と検証用画像の損失を見ながら適切な epoch 数を選べば Over fittingを防げると考えて、今回は dropoutの機能は使用していない。出力の次元、つまり教師データは4個とした。どのような情報を教師データとしたかについては学習用画像の節で述べる。

参考文献 [1] のサンプルソースコードからの変更点は、出力結果の教師データへの分類から、連続値である教師データの予測 (回帰) に変更するために、最終段の分類の為に挿入されている SoftmaxWithLoss の層を、出力をそのまま出力し損失を計算する、自作の IdentityFunctionWithLoss の層に変更した点である。この分類から回帰への変更に伴い、損失関数を「交差エントロピー誤差」から「2乗和誤差」に変更した。最適化の為に Optimizer には、現在最も使われていると思われる Adam を使用した。この21層のニューラルネットワークを使用するに当たっては、参考文献 [1] の他に参考文献 [2,3] が非常に参考になった。

\* masahiro.nomura@j-parc.jp

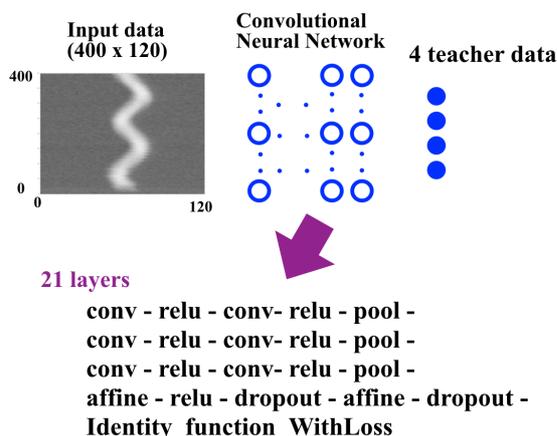


Figure 1: Network configuration diagram.

### 3. 学習用画像

学習用画像は教師データが正確に分かっている画像を数千枚以上揃える必要がある。このような多くの画像を、実際の画像で揃えることは困難であり、学習用画像はシミュレーションにより作成した。その為、実測したマウンテンプロットにシミュレーションにより作成した画像を近づけることが重要である。実際の画像とシミュレーションとの一番違いは、バックグラウンドの存在である。そこで、先ず実際のマウンテンプロットのバックグラウンドの状態を調べ、シミュレーションに反映するように様にした。

#### 3.1 マウンテンプロット

先に述べた様に、マウンテンプロットとは、RCS内に設置された Wall Current Monitor の波形を周回毎にスライスし、縦に並べて表示した画像である。縦軸が Slice number を表し、横軸は時間を表している。図の高さ方向は粒子の強度分布を表している。Wall Current Monitor では 5[nsec] 毎にデータを取得している。実際のマウンテンプロットの例を Figure 2 に示す。画像は gnuplot により作成している。

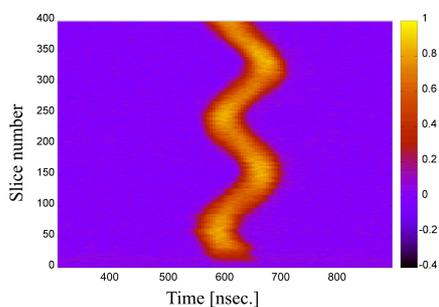


Figure 2: Example of mountain plot.

現状では、このカラー画像から RCS への入射タイミングと入射運動量、正確には RCS の RF を基準とした設定値からのオフセットを人がこの画像を実

際に測ることで求めている。今回の画像認識では、強度分布を読み取りやすくするために、このようなカラー画像では無く、単色の白黒画像を使用した。さらに、ビーム強度依存性を無くす為に、ビーム強度の最大値を 1 に規格化した画像を画像認識用に使用した。このような措置を行なった画像の例を Figure 3 に示す。また、Figure 3 には Slice number = 100 での histogram も示す。

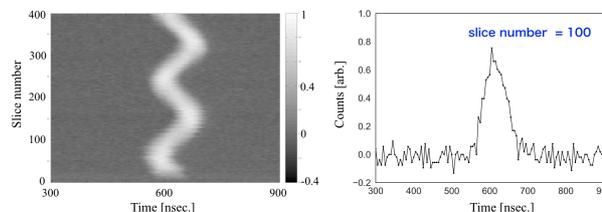


Figure 3: Mountain plot for image recognition and histogram at slice number = 100.

Figure 3 の histogram から分かる様に、全対的にバックグラウンドがあることが分かる。バックグラウンドの状況を調べるために、両端 100[nsec.] 程度の領域でのその大きさを調べた。結果を Figure 4 に示す。赤線は Gaussian での fitting を表している。平均  $\mu = -0.03$ 、 $\sigma = 0.06$  であった。平均  $\mu$  がほぼ 0 であることから、バックグラウンドレベルはほとんど 0 であり、平均的には、Figure 3 の histogram での高さへの影響はほとんど無いと考えられる。バックグラウンドの影響はビームの端の部分に現れると考えられる。画像認識を行なう画像は、大きさ 0.2 以下をバックグラウンドと考えて 0 に設定した画像を使用した。シミュレーションで学習用画像を作成する際は、これらの情報を元に、バックグラウンドを含んだ画像を作ることにした。

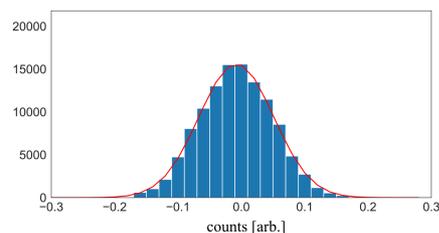


Figure 4: Histogram of background counts.

#### 3.2 シミュレーションによる学習用画像の作成

シミュレーションコードは、シンクロトロン方程式から、位相空間での運動を計算しただけの単純なもので、空間電荷効果は含んでいない。シミュレーションに使用した主なビームの条件は、RCS への入射 energy = 400 [MeV]、出射 energy = 3 [GeV]、マルチターン入射数 = 30 であり、加速電圧のパターンは現在使用しているパターンを使用した。教師データ

の情報を含んだ多量の画像を作るために、教師データは乱数で発生させた。教師データとしては、入射運動量オフセット、入射タイミングオフセット、入射ビーム運動量広がり、入射ビーム時間幅とした。教師データの示すビーム条件の概略図を Figure 5 に示す。オフセットとは、RCS の RF を基準とした設定値からのズレを表している。先に述べたように、実際の画像に近づけるため、シミュレーションした粒子軌道に Gaussian( $\mu=0, \sigma=0.06$ ) で作成したノイズを加えて、0.02 以下をカットした画像を作成した。今回、学習用画像数 8000 枚、検証用画像 2000 枚作成した。

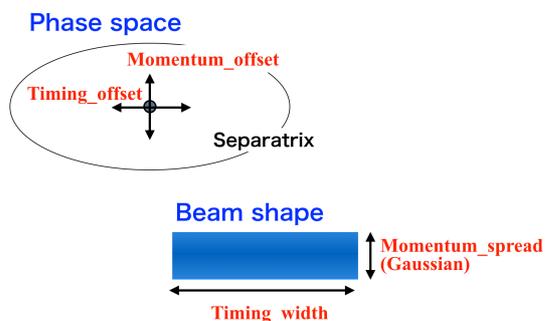


Figure 5: Schematic view of teacher data.

#### 4. 学習と検証

今回画像から求めた情報は、調整時に人が実際に画像を測って求めている情報、入射運動量オフセットと入射タイミングオフセット、それと、マルチターン入射のマウンテンプロットからでは難しい運動量広がりについてである。学習は batch size = 40、epochs = 60 の設定で行なった。epoch とは、決められた batch size で全ての与えられた学習用画像数を処理した場合の繰り返しを表す単位で、1 回 40 画像のバッチ学習を  $8000/40 = 200$  回行なったことに対応している。今回は、この学習を 60 回繰り返したことになる。学習用画像と検証用画像の損失の epoch 依存性を Figure 6 に示す。検証用画像の損失が学習用画像の損失と乖離して大きくなっていないことから、この epoch の範囲では、Over fitting にはなっていないことが分かる。

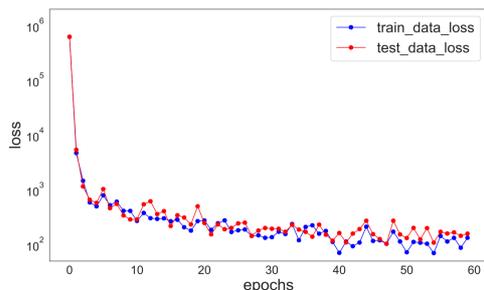


Figure 6: Loss of training data and verification data.

学習の結果を確かめる為に、検証用画像 20 枚を選んで、教師データと予測値の比較を行なった。結果を Figure 7 に示す。CNN による予測値は、学習していない検証用のマウンテンプロットの画像を正しく認識することにより、教師データを正しく再現していることが分かる。

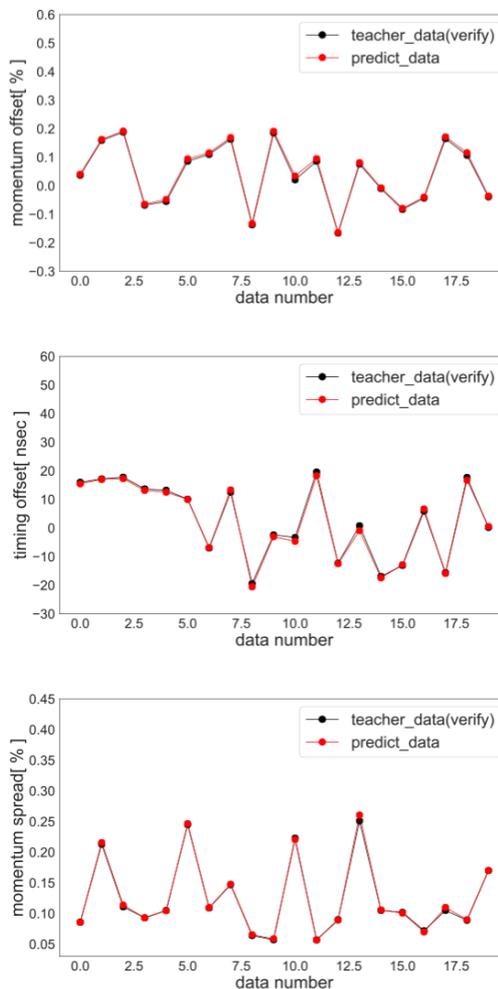


Figure 7: Comparison of teacher data and prediction data.

#### 5. 実際の画像への適用

まず初めに、RCS の調整時に使われる入射運動量オフセットと入射タイミングオフセットを求めた結果について述べる。Figure 8 は、ある日の調整の様子を表した図で、入射運動量オフセットを調整し、上から下にかけて段階的に調整している様子を表している。左側が実測のマウンテンプロット、右側が CNN による予測値から作成したマウンテンプロット、つまり右側は、CNN が左側のマウンテンプロットをこの様な画像だと認識した画像である。両者を比較すると、ほぼ同様の形状をしており、測定したマウンテンプロットから CNN が正しく画像を認識し

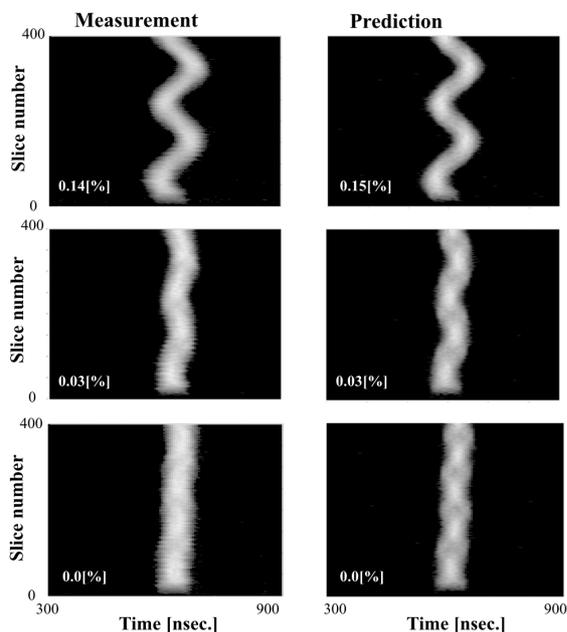


Figure 8: Measurement and prediction images.

情報を求めていることが分かる。図中の数字は人が画像から測定した運動量オフセットと CNN が予測した運動量オフセットであり、ほぼ同様の数値を示している。この他にいくつかの画像からも入射運動量オフセットと入射タイミングオフセットを求めた結果、人が画像から読み取った値と画像認識により求めた値の違いは、運動量オフセットが 0.02[%]、タイミングオフセットが 4[nsec.] 程度であった。CNN の予測した画像が実際の画像とほぼ同じことから、この画像認識の技術は RCS への入射の運動量やタイミングの調整に使える段階にあり、画像認識により正しく情報を得ることができれば、効率よく一度で設定値に調整できる可能性があると考えられる。

次に、マルチターン入射のマウンテンプロットの画像からでは難しい運動量広がりをもとめた結果について述べる。Figure 9 は運動量オフセットとタイミングオフセットを調整した後、Debuncher2 を ON にして運動量を広げた様子を示している。左側の図が Debuncher2 OFF、右側の図が Debuncher2 ON を表している。

運動量広がり数値については、1 中間バンチ入射に条件を変更し、トモグラフィーによる測定結果では、通常 Debuncher2 ON と OFF の状態では、

$$\Delta P/P = 0.10[\%] \quad (\text{Debuncher2 OFF}) \quad (1)$$

$$\Delta P/P = 0.17[\%] \quad (\text{Debuncher2 ON}) \quad (2)$$

である。一方、このふたつの実測のマウンテンプロットから CNN による画像認識により求めた運動量広がり、

$$\Delta P/P = 0.09[\%] \quad (\text{Debuncher2 OFF}) \quad (3)$$

$$\Delta P/P = 0.16[\%] \quad (\text{Debuncher2 ON}) \quad (4)$$

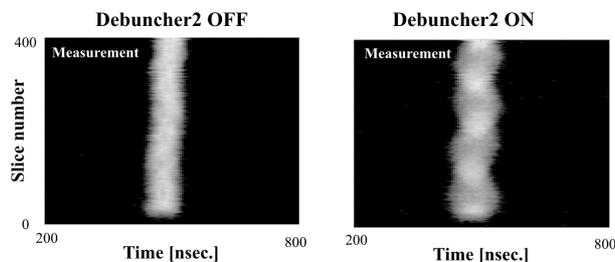


Figure 9: Increasing of momentum spread with Debuncher2.

であった。

測定回数増やし、トモグラフィーによる測定結果と数多く比較する必要はあるが、マルチターン入射のマウンテンプロットからでも、このように運動量広がり求められると考えられる。

## 6. まとめ

今回初めて CNN を用いた画像認識の技術のマウンテンプロットに適用し、ビームに関する情報を求めた。その結果、入射運動量と入射タイミングオフセット、さらには運動量広がりについてもマウンテンプロットから求めることができると考えられる。

この画像認識の技術を用いることにより、入射運動量と入射タイミングオフセットについては、「効率よく一度で設定値に調整できる可能性がある」、「誰でも調整が可能になる」のメリットがあり、運動量広がりについては、測定回数を増やし、確かめていく必要があるが、「測定条件的には難しい、マルチターン入射のまま簡単に測定できる。」といったメリットがある。これらのことから、この CNN による画像認識の技術は今後実際に使っていくべき技術であると考えられる。

## 参考文献

- [1] 斎藤康毅, ゼロから作る Deep Learning  
—Python で学ぶディープラーニングの理論と実装,  
オライリー・ジャパン, オーム社,  
ISBN978-4-87311-758-4.
- [2] 21 層のネットワークを動かす 1; <https://1x2x3.tech/deep-learning/21-layers-network-001>
- [3] 21 層のネットワークを動かす 2; <https://1x2x3.tech/deep-learning/21-layers-network-002>