深層学習におけるデータ拡張による汎化性能の向上に関する研究

情報数理応用研究　5218C031-1　藤波英輝
指導教員　後藤正幸

Research on Improvement of Generalization Performance by Data Augmentation in Deep Learning

HIDEKI Fujianmi

1. はじめに

深層学習は画像認識 [1],[2] などの分野において成功を納め，人工知能ブームの火付け役となった．しかし，深層学習は，モデルに多くのパラメータが使われているため，学習データが有限の状況では，少数の学習データの特徴のみを過度に捉えてしまい汎化能力が低下するという問題がある．一般に，画像データなどに正解のカテゴリラベルを付与するには人間による判断が必要であり，大量の学習データを獲得するには多くの人的コストがかかる．そこで，データ拡張手法 [1] により学習データを増やし，モデルの汎化性能を向上させるアプローチがある．

深層学習を用いた画像認識を対象としたモデルにおけるデータ拡張手法には，horizontal flipping など画像の一部分に対して回転などの変換を加えるような手法が多数提案されている [1],[3]．しかし，これらの手法はデータ構造に合わせて設計されているため，画像データ以外に対して応用することができない．加えて，データ拡張手法による汎化能力の向上の程度は，画像データのドメインの違いに依存する．例えば，horizontal flipping は物体認識においては有効なデータ拡張手法であるが，文字認識においてはあるクラスの画像を対称に変化させた場合，他のクラスの画像と類似してしまうため，変換した画像が元の画像と同じクラスであるという仮定を満たさなくなり，元の学習データに対して under-fitting が起こる．

一方，Generative Adversarial Network(GAN)[4] に基づいたデータ拡張手法が近年研究されている．GAN は識別モデルと生成モデルを敵対的に学習することで，生成モデルによって生成されるサンプルの分布を学習データの経験分布に近似する．しかし，GAN では多クラスのデータを学習しても，特定のクラスを指定してデータ生成することができない．そのため，GAN を特定のクラスを指定して画像データを生成することが可能となるように拡張した Conditional Generative Adversarial Networks(CGAN)[5] がデータ拡張手法として用いられる．しかし，分類モデルの学習データが十分ではない場合は，CGAN の学習にも十分なデータが与えられないことになる．このとき，学習済みの CGAN によって，生成される画像は入力したクラスラベルに対応するクラスの特徴を捉えることができない．そのため，生成された画像がノイズ（外れ値）となってしまう．このような場合，データ拡張された追加データが学習を阻害し，元々存在した学習データに対して under-fitting が起こる．

また，CGAN において学習データの経験分布が近似できているとき，生成モデルからランダムにサンプリングされるデータは特徴空間においてデータが密に存在する領域に生成されることになる．そのため，生成したデータを学習データとして追加しても，データセットの多様性が向上せず，学習データに対する over-fitting を回避することができない．

そこで，本研究では上記の 2 つの課題（under-fitting と over-fitting）を解決するデータ拡張手法を提案する．まず，元の学習データに対する under-fitting を避けるため，生成したデータがノイズとらない方法を考える．生成データのノイズ化は，学習データ数に対して，生成対象データの次元数が大きいために起こる．そこで，本研究では，CGAN の代わりに，各クラスごとに独立に学習した GAN によって，新たな学習データを生成する．これによって，CGAN におけるラベル情報の入力を回避し，生成データの次元数を削減する．加えて，学習データに対する over-fitting を回避するために，学習データと生成したデータの位相関係に着目する．元の学習データが密に分布している領域に生成されたデータは，over-fitting を促す．これに対し，元の学習データが疎な領域に生成されかつ分類境界付近のデータを学習することでモデルの over-fitting を改善し，汎化性能の向上が見込める．そこで，分類境界付近に新たな学習データを生成する方法を提案し，この手法で生成されたデータを学習することで，over-fitting を回避する．本提案モデルではこのように，元の学習データに対する under-fitting と over-fitting を回避し，モデルの汎化性能を向上させる．そして，画像分類のベンチマークデータセットによって，従来手法と提案手法の分類精度を比較することで，提案手法の有効性を確認する．

2. 関連研究

画像分類のための深層学習モデルに用いられるデータ拡張手法には主に，元の学習データに対して回転などの変換を加える方法と，学習データを新たに生成する方法が存在する．元の学習データに対して回転などを加える方法には horizontal flipping や, color shifting などがある [1]．これらの手法は「元の画像に対する微小変化を加えたデータは，元のデータと同じクラスのデータである」という前提に基づいている．その仮定のもとで，データに回転や色彩変化などの画像というデータ構造に合わせた変換を加えたデータを生成し，新たな学習データとして追加する．このような手法は対象となる画像データセットごとに適切な変換方法が定まっているわけではなく，実験者がどのデータ拡張手法によって汎化性能が向上するのか検証して確認する必要がある．

学習データを新たに生成する手法としては，深層学習による生成モデルを用いたデータ拡張手法が提案されている [6]．この研究では，CGAN により生成した新たなデータを学習デー

タとして活用しており，CGAN によって生成されたデータが分類モデルの識別性能を向上させるための 2 つの要件を提示している．1 つ目は，特徴空間において生成画像群が分散している点である．この要件を満たしていないとき，CGAN は学習データのうち生成しやすい特定の学習データのみを生成していることになる．そのため，生成されたデータ自体の多様性が小さく，分類モデルの学習データに追加すると，これらの学習データに対して過学習してしまう．2 つ目は生成画像群が入力空間において学習データが分布している領域から外れている点である．これにより，生成されるデータが学習データと異なる特徴と共通する特徴の両方を内包するため，学習データに対する over-fitting を抑制することができる．この手法 [6] では以上の 2 つの要件を満たすように CGAN を学習する方法を提案しているが，PixelCNN++[7] という計算負荷の高い手法によって，画像の事前確率分布を求める必要があり，計算時間の観点から現実的なモデルとなっていない．

3. 準備

3.1. GAN

GAN は学習データの画像を再現できる生成モデルとして脚光を浴びている．GAN では，生成モデル $G(\boldsymbol{z})$ と識別モデル $D(\boldsymbol{x})$ を敵対的に学習する．生成モデルが識別モデルに誤分類させるように学習することで，生成モデルが生成するサンプルの分布を学習データの経験分布に近似する．GAN では以下の式 (1) を最適化するように生成モデル，識別モデル双方のパラメータを学習することで，生成した画像の分布と学習画像データの経験分布の JS ダイバージェンスを最小化している．

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim\hat{p}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z})))] \tag{1}$$

ここで，V，D，G はそれぞれ目的関数，識別モデル，生成モデルを表し，$\boldsymbol{x}$ は学習データの経験分布 $\hat{p}$ からサンプリングされたデータ，$\boldsymbol{z}$ はノイズ分布 $p_{\boldsymbol{z}}$（標準正規分布）からサンプリングされたノイズデータを表す．

3.2. CGAN

通常の GAN は多クラスのデータで GAN を学習したとき，特定のクラスのデータを意図的に生成することができない．そのため，多クラスのデータで学習した単一の GAN によってデータ拡張をする場合，生成された画像に対して手動でラベルをつける必要がある．これに対して CGAN では，多クラスのデータで学習し，特定のクラスの画像を意図的に生成することが可能である．しかし，CGAN では識別モデルの学習において，画像にラベル情報として，クラス数分のチャネルを追加する必要がある．そのため，GAN と比較したとき，入力される学習データの次元数が大きくなってしまい，学習データが少ない場合には，過学習により，ノイズ画像が生成されてしまう．また，CGAN の学習が収束し，学習データの経験分布が近似されたとき，CGAN は元の学習データが密に分布している領域にデータを生成する傾向を持つ．そのため，学習済みの CGAN によって生成されたサンプルは学習データと類似した特徴を持ち，分類モデルの学習データとして増強しても学習データの多様性を増加させない可能性が高い．このとき，学習データに対する over-fitting を促してしまうため，汎化性能が低下してしまう．CGAN では以下の目的関数の最適化を行なっている．

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim\hat{p}}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \tag{2}$$

ここで，y は学習データのクラスラベルを表す．

4. 提案モデル

CGAN をデータ拡張に用いるために少数データで学習する場合，ノイズ画像が生成されると under-fitting，生成された学習データが元の学習データの over-fitting につながり，分類モデルが汎化性能の向上しない．そこで本研究では，以下の 2 点に着目したデータ拡張手法によって，学習データに対する under-fitting と over-fitting を回避する．

1. 生成モデル（CGAN）におけるパラメータ数を削減することで，学習データが少ない場合でも，ノイズ画像が生成されることを回避する．

2. 分類モデルの元の学習データに対する過学習を回避するために，元の学習データがスパースな領域かつ分類境界付近の生成データを分類モデルの新たな学習データとして追加する．

まず 1 点目の着眼点を満たすために，CGAN の代わりに，分類モデルの学習データのクラスごとに，独立に GAN を学習する．これによって，CGAN における，クラス情報を学習に取り込むためにパラメータ数が増加する問題を回避し，パラメータ数を削減することができる．次に 2 点目の着眼点を満たすために，分類モデルの出力確率のエントロピーに着目する．深層学習では，入力空間において，類似した特徴を有するクラスの分類においても，分類に必要な抽象表現を学習することが可能である．そのため，元の学習データは特徴空間において分類モデルの特定のクラスの出力確率が高い領域に密に存在し，逆にどの出力確率も平均的な領域では元の学習データは疎となると考えられる．

そこで，出力確率分布のエントロピーが大きく，学習データが疎な領域に生成されたデータを新たな学習データとして，元の学習データに追加して学習する．これによって，学習データの多様性を改善し，元の学習データに対する over-fitting を抑制することで，分類モデルの汎化性能向上を実現する．出力確率分布のエントロピーは以下の式 (3) によって求める．

$$w(\boldsymbol{z}) = -\sum_{j=1}^{M} p_j(G(\boldsymbol{z})) \log p_j(G(\boldsymbol{z})) \tag{3}$$

ノイズ $\boldsymbol{z}$ を入力して得られる生成画像 $G(\boldsymbol{z})$ を分類モデルに入力したき，クラス j に分類される確率を $p_j(G(\boldsymbol{z}))$ と定義する．また $j \in \{1, 2, \cdots, M\}$ とし，M はクラス数を表す．上記のエントロピーを用いたデータ拡張による分類モ

デルの学習アルゴリズムを以下に示す．ここで，分類モデル，GAN の学習回数（エポック）をそれぞれ $N_{\text{classifier}}$，N_{gan} とし，分類モデルをデータ拡張して再学習する際の学習回数を N_{aug} とする．またデータ拡張では各クラスごとに N_s 回の生成を行い，そのうちエントロピーの大きいデータを追加データ集合として，その経験分布を $p_{aug}(\boldsymbol{x}, \boldsymbol{y})$ と定義する．分類モデル，GAN の識別モデル，生成モデルそれぞれのパラメータは $\boldsymbol{\theta}_C, \boldsymbol{\theta}_{D_j}, \boldsymbol{\theta}_{G_j}$ と定義し，学習データ集合の経験分布として近似される $\hat{p}(\boldsymbol{x}, \boldsymbol{y})$ または $\hat{p}(\boldsymbol{x})$ からランダムにサンプリングされたバッチサイズ m の学習データによって確率的勾配降下法により学習する．このとき，パラメータの更新方法には Adam を用い更新量はそれぞれ $\alpha_{\boldsymbol{\theta}_C}, \alpha_{\boldsymbol{\theta}_{Dj}}, \alpha_{\boldsymbol{\theta}_{Gj}}$ と定義する．

（元の学習データのみで分類器を学習）

for $i = 1, ..., N_{\text{classifier}}$ **do**
  $\{(\boldsymbol{x}_i^{(1)}, \boldsymbol{y}_i^{(1)}), ..., (\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)})\} \sim \hat{p}(\boldsymbol{x}, \boldsymbol{y})$
  $Loss_{train} = \frac{1}{m} \sum_{k=1}^{m} L(\boldsymbol{\theta}_c, \boldsymbol{x}_i^{(k)}, \boldsymbol{y}_i^{(k)})$
  $grad_C = \nabla_{\boldsymbol{\theta}_C} Loss_{train}$
  $\boldsymbol{\theta}_C \longleftarrow \text{AdamUpdate}(\alpha_{\boldsymbol{\theta}_C}, \boldsymbol{\theta}_C, \boldsymbol{grad}_C)$
end for

（各クラスごとに GAN を学習）

for $j = 1, ..., M$ **do**
  for $i = 1, ..., N_{\text{gan}}$ **do**
    $\{\boldsymbol{z}_i^{(1)}, ..., \boldsymbol{z}_i^{(m)}\} \sim p_z(\boldsymbol{z})$
    $\{\boldsymbol{x}_i^{(1)}, ..., \boldsymbol{x}_i^{(m)}\} \sim \hat{p}(\boldsymbol{x})$
    $\boldsymbol{grad}_{D_j} = \nabla_{\boldsymbol{\theta}_{D_j}} \frac{1}{m} \sum_{k=1}^{m} [\log D_j(\boldsymbol{x}_i^{(k)}) + \log(1 - D_j(G_j(\boldsymbol{z}_i^{(k)})))]$
    $\boldsymbol{\theta}_{D_j} \longleftarrow \text{AdamUpdate}(\alpha_{\boldsymbol{\theta}_{D_j}}, \boldsymbol{\theta}_{D_j}, \boldsymbol{grad}_{D_j})$
    $\boldsymbol{grad}_{Gj} = \nabla_{\boldsymbol{\theta}_{G_j}} \frac{1}{m} \sum_{k=1}^{m} \log(1 - D_j(G_j(\boldsymbol{z}_i^{(k)})))$
    $\boldsymbol{\theta}_{G_j} \longleftarrow \text{AdamUpdate}(\alpha_{\boldsymbol{\theta}_{G_j}}, \boldsymbol{\theta}_{G_j}, \boldsymbol{grad}_{G_j})$
  end for
end for

（エントロピーの高いデータを新たな学習データとして追加）

$\{g, w\} = NULL$（追加生成データとそのエントロピー）
for $j = 1, ..., M$ **do**
  for $i = 1, ..., N_{\text{s}}$ **do**
    $\{\boldsymbol{z}_i^{(1)}, ..., \boldsymbol{z}_i^{(m)}\} \sim p_z(\boldsymbol{z})$
    for $k = 1, ..., m$ **do**
      $w(\boldsymbol{z}_i^{(k)}) = -\sum_{j=1}^{M} p_j(G_j(\boldsymbol{z}_i^{(k)})) \log p_j(G_j(\boldsymbol{z}_i^{(k)}))$
      $\text{Stack}(\{g, w\}, \{G_j(\boldsymbol{z}_i^{(k)}), w(\boldsymbol{z}_i^{(k)})\})$
    end for
  end for
end for
$p_{aug}(\boldsymbol{x}, \boldsymbol{y}) = \text{SortDrop}(\{g, w\})$

（生成データと元の学習データで分類器を学習）

for $i = 1, ..., N_{\text{aug}}$ **do**
  $\{(\boldsymbol{x_{aug}}_{,i}^{(1)}, \boldsymbol{y_{aug}}_{,i}^{(1)}), ..., (\boldsymbol{x_{aug}}_{,i}^{(m)}, \boldsymbol{y_{aug}}_{,i}^{(m)})\} \sim p_{\text{aug}}(\boldsymbol{x}, \boldsymbol{y})$
  $\{(\boldsymbol{x}_i^{(1)}, \boldsymbol{y}_i^{(1)}), ..., (\boldsymbol{x}_i^{(m)}, \boldsymbol{y}_i^{(m)})\} \sim \hat{p}(\boldsymbol{x}, \boldsymbol{y})$
  $Loss_{train} = \frac{1}{m} \sum_{k=1}^{m} L(\boldsymbol{\theta}_c, \boldsymbol{x}_i^{(k)}, \boldsymbol{y}_i^{(k)})$
  $Loss_{aug} = \frac{1}{m} \sum_{k=1}^{m} L(\boldsymbol{\theta}_c, \boldsymbol{x_{aug}}_{,i}^{(k)}, \boldsymbol{y_{aug}}_{,i}^{(k)})$
  $grad_C = \nabla_{\boldsymbol{\theta}_C}(Loss_{train} + Loss_{aug})$
  $\boldsymbol{\theta}_C \longleftarrow \text{AdamUpdate}(\alpha_{\boldsymbol{\theta}_C}, \boldsymbol{\theta}_C, \boldsymbol{grad}_C)$
end for

上記のアルゴリズムでは，先に元の学習データのみで分類モデルを学習することによって（1〜6 行目），生成したデータを分類モデルの出力確率のエントロピーで評価する際の，分類モデルの評価結果の頑健性を担保する．次に各クラスごとに学習データから独立に GAN を学習する（7〜16 行目）．そして学習済みの各クラスの GAN によって生成された画像を分類モデルに入力して，出力確率のエントロピーを計算し（17〜26 行目），学習データに追加する生成データのみをサンプリングする（27 行目）．そして，最後に，追加された学習データと元の学習データによって，分類モデルを再学習する（28〜35 行目）．このとき，元の学習データとともに再学習することで，生成したデータに対する過学習を抑制することができる．

5. 評価実験

Mnist, CIFAR10 を用いた 10 クラス分類の評価実験を行い，提案手法と従来手法の分類精度を比較する．従来手法としてはデータ拡張を用いず学習したモデル，CGAN によってデータ拡張して学習したモデル，Manifold Mixup[8] によるデータ拡張したモデルを用いる．このとき，分類モデルの構造としては WideResnet28[9] を用い， 各生成モデル (GAN，CGAN)，分類モデルについてはエポック数 $N_{classifier} = 300$，全てのモデルにおいてバッチサイズは $m = 32$ で学習を行った．各モデルのパラメータの更新量は GAN の生成モデルと識別モデルは $\alpha_{\boldsymbol{\theta}_D} = \alpha_{\boldsymbol{\theta}_G} = 0.0002$，分類モデルのパラメータの更新量は $\alpha_{\boldsymbol{\theta}_C} = 0.01$ とする．また各クラスごとに学習に用いる教師データの数は 400 枚（合計で 4,000 枚），テストデータはクラスごとに 1,000 枚 (合計 10,000 枚) とする．また学習として用いられる 4,000 枚のデータの違いによる分類精度の差を緩和するために，テストデータ以外のデータから学習データを 10 回サンプリングし直し，その度に測定したテストデータに対する分類精度を求めた．その平均を以下の表 1 に示す．

表 1: 10 クラス分類の分類精度の比較（%）

分類対象 データセット	提案 モデル	CGAN	Manifold Mixup	データ拡張 なし
Mnist	**99.289**	85.321	97.876	97.879
Cifar10	**75.722**	60.773	62.001	60.661

実験結果から，本提案アルゴリズムではデータ拡張を行わないで分類モデルを学習した場合と比較して，高い汎化性能を実現していることがわかる．また，CGAN によって生成されたデータによってデータ拡張して，分類モデルを学習したときの結果と比較しても，分類精度の向上が確認できた．また多クラスのデータで学習した CGAN によって生成したデータを用いて分類モデルの学習を行う方法は，データ拡張しなかった場合と比較して分類精度が悪化している．CGAN によって生成したデータを用いたデータ拡張では，ノイズと

なる画像が生成されてしまうことで，元の学習データに対して under-fitting が起こり，データ拡張なしで学習したモデルよりも汎化能力が悪化したと考えられる．一方，提案モデルでは，ノイズ画像の生成を回避した上で，多様な学習データによって学習したことで，データ拡張しない場合よりも，分類精度が向上したと考えられる．続いて，提案モデルによるデータ拡張の効果と追加するデータ数の関係についての関係を明確にするために，学習データ数を固定して，追加するデータ数を変化させたときの，テストデータの分類精度を比較する（表 2）．このとき，元の学習データ数は各クラスごとに 400 枚（合計で 4,000 枚）とする．

表 2: 追加する学習データ数と分類精度（%）

追加する学習データ数	400	800	1,200	1,600	2,000
分類精度	75.722	75.635	75.822	75.000	75.861

実験結果から，追加する学習データ数がもとの学習データに対して大幅に増えたとしても，分類精度の向上は確認できなかった．これは，追加する学習データ数が一定以上となると，分類モデルにおけるエントロピーを大きくする追加データが識別精度を向上させないノイズ画像になってしまうためと考えらえる．

また，提案手法によって追加された学習データのうち最もエントロピーの大きな Mnist 画像の例を図 1 に示す．

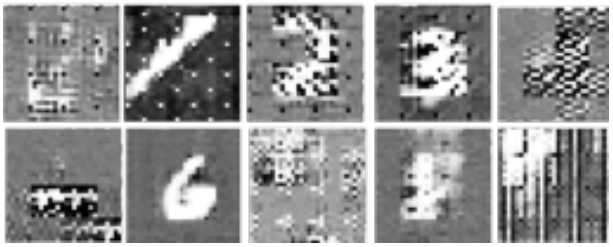

図 1: 提案モデルにおける Mnist の生成画像

生成画像を確認すると，クラスごとにノイズの大きさが異なることが分かる．“2”,“3”,“6” の画像に対して，“5”,“7”,“9” はノイズ画像となっている．この結果から，クラスごとに生成のしやすさに差があり，データ拡張の効果がクラスごとに異なると考えられる．

6. 考察

2 種類のベンチマークデータセットによる分類実験の結果から，本提案手法によって，従来手法よりも高い汎化性能を示すことができた．これは提案手法のデータ拡張で追加された学習データによって，元の少数の学習データに対する過学習を回避し，元の学習データを学習しただけでは，誤分類しやすいデータを識別可能になったためと考えられる．他方，学習データに対して，追加する学習データを増加させたとき，汎化性能の向上は確認できなかった．これは，生成されたデータが一定以上ある場合は，追加された学習データと元の学習データの経験分布がほとんど変化せず，追加された学習データによって，元の学習データの多様性が向上されなくなるためだと考えられる．

また，提案手法によって追加された画像を確認すると，クラスごとに生成した画像が実際のクラスの特徴を捉えているものとそうでないものがあった．このことから，生成された画像において，単にエントロピーが大きなデータのみをサンプリングすると，追加する学習データにノイズ画像が含まれてしまう．これによって，分類モデルの学習が阻害されることが考えれられるので，エントロピーの大きなデータのうち，ノイズ画像をリジェクトすることで，分類精度をさらに向上させることが可能であると考えられる．

7. まとめと今後の課題

本研究では，GAN により，学習データの多様性を増強し，under-fitting することなく汎化性能を向上させるデータ拡張法を提案した．またベンチマークデータセットに対する分類実験を行い，テストデータの分類精度を比較することで，提案手法の有効性を示した．

今後の課題としては，提案手法では，生成したデータのうち学習に追加するデータをエントロピーで評価しているため，生成したデータにノイズがあった場合に，そのデータをリジェクトする方法が挙げられる．そのためのノイズ画像の評価方法について検討する必要がある．

参考文献

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks,” *In Advances in Neural Information Processing Systems*, 2012.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence*, 37(9),pp. 1904–1916, 2015.

[3] A.G. Howard.“Some improvements on deep convolutional neural network based image classification,” *The computing Research Repositry*, abs/1312.5402, 2013.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *In Advances in Neural Information Processing Systems*, 2014.

[5] M. Mirza and S. Osindero, Conditional generative adversarial nets, *The computing Research Repositry* abs/1411.1784, 2014.

[6] Z. Dai, Z. Yang, F. Yang, W. W. Cohen and R. R. Salakhutdinov, “Good semi supervised learning that requires a bad gan,” *Advances in Neural Information Processing Systems*, pp. 6513–6523, 2017.

[7] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, “Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications,” *In Proceedings of the International Conference on Learning Representations* Poster, 2017.

[8] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio,“Manifold mixup: Better representations by interpolating hidden states,” *In International Conference on Machine Learning*, pp.6438–6447, 2019.

[9] S. Zagoruyko and N. Komodakis,“Wide residual networks,” *In British Machine Vision Conference*, 2016.