卒　業　論　文
Graduation Thesis

題　　目　学習サンプルが少ない状況下での Vision Transformer の性能評価に関する研究

TITLE: Study on performance evaluation of Vision Transformer under small training samples

指導教員　水上 嘉樹　教授
SUPERVISOR: Associate Prof. Yoshiki MIZUKAMI

提 出 者　佐藤 立樹
AUTHOR: Tatsuki SATO

提出年月　令和 6 年 2 月
DATE: February, 2024

山口大学工学部知能情報工学科
Department of Information Science & Engineering,
Faculty of Engineering, Yamaguchi University

要約：

本研究では，学習サンプルが少ない状況下での Vision Transformer の精度について研究する．比較対象として ResNet，EfficientNet を使用する．その際，事前学習をしていた場合とそうでない場合で実験を行う．学習サンプル数として 1000，100，50，20 を用いた場合の４パターンである．

本研究の比較によって，事前学習をしていなかった場合はどのモデルも振るわない結果となり，事前学習をしていた場合の Vision Transformer が最も有望であることが確認された．

Abustract:

In this study, we study the accuracy of the Vision Transformer under a small training sample. ResNet and EfficientNet will be used for comparison. Experiments are conducted with and without prior training. The training data are 1000, 100, 50, and 20 patterns.

The comparisons in this study confirm that none of the models perform well when no pre-training is performed, and that the Vision Transformer is the most promising when pre-training is performed.

目次

第 1 章　本研究の背景と目的

1.1　背景

近年人工知能が急激に発達し，今までは人が手動で行っていた作業などが次々と人工知能が担うようになってきている．その中には，X 線の画像など，医療関係の画像の識別も人工知能に頼っている部分があり，自動車の自動運転も開発が行われていることから，画像認識の技術が急速に進展してきていることがわかる．これは技術が発展している証拠であるため良いことではあるが，逆に画像認識の精度がかなり高くないと危険が及んでしまうので，失敗は許されないとも考えることができる．

画像認識技術の中でも，近年提案された Vision Transformer はその高い認識精度が注目されている[5]．もともと Vision Transformer は学習サンプルが数億枚という数に達した時に真価を発揮するという特性を持っている．しかし数億枚という数を集めるには莫大な時間や労力がかかる場合があるため，そこまでの数を集めるのは現実的に困難であるといった状況は多々あるはずである．そこで，学習サンプルが少ない状況下での Vision Transformer の精度は高い精度を発揮するのか，他の画像認識技術と比較してどれほどの優劣があるのかを調査する必要がある．

1.2　目的

本研究では，Vision Transformer を用いて犬と猫の判定を行う．犬と猫のデータセットを用いて学習を行う際，学習サンプルの数を少数にする．同様の実験を ResNet および EfficientNet を用いて行い，Vision Transformer との精度を比較することを目的とする．
その際，事前学習を有りにした場合と無しにした場合とも比較を行う．

第 2 章　関連研究

2.1　従来の画像認識技術との相違点

従来の画像認識技術として，「CNN」といった，「畳み込み」という操作を加えたニューラルネットワーク構造が挙げられる．畳み込みとは，比較的サイズの小さい格子上の数値データ（フィルター）と，同じサイズの部分画像（ウィンドウ）を計算させることで 1 つの値が求められる．それを格納し，フィルターをずらして再度計算させる．これを全ての入力に対して計算し，得られた数値の集合が，局所的に抽出された特徴量として出力される．これらの様子を図 2.1 に示す．

一方 Vision Transformer は，入力画像と同じサイズの線形フィルタであり，その入力画像によって Attention Matrix と呼ばれる線形フィルタの重みを変えるというものである．これによって，画像全体から局所的な特徴だけでなく，大きな特徴や局所的な特徴の組み合わせを得ることができる．これが Vision Transformer の主な特徴であり，従来の画像認識技術との相違点である．

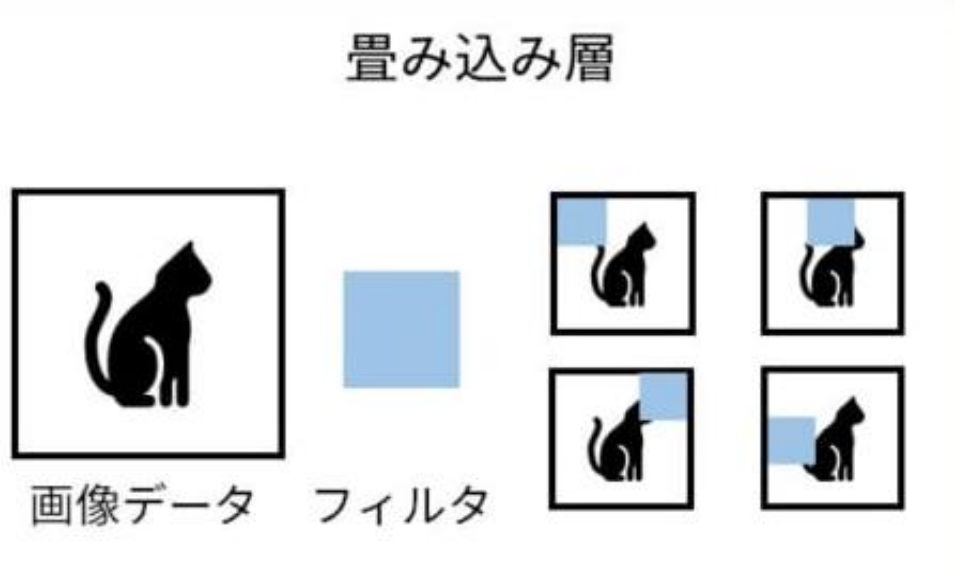


図 2.1 畳み込み例
出典 [1]より引用

2.2　Vision Transformer の構成

本研究で用いる Vision Transformer について述べる．Vision Transformer は，入力画像を決まった数の画像（パッチ）に分割し，それらを並べて次節で述べる Transformer Encoder に入力される．その際，パッチの位置を定めるために Position Embedding と呼ばれるトークンをそれぞれのパッチに挿入する．これにより，前節でも述べたように画像全体から局所的な特徴だけでなく，大きな特徴や局所的な特徴の組み合わせを得ることが可能となる．その後処理を行い，抽出された特徴により画像を分析し出力される．これらの様子を図 2.2 に示す．

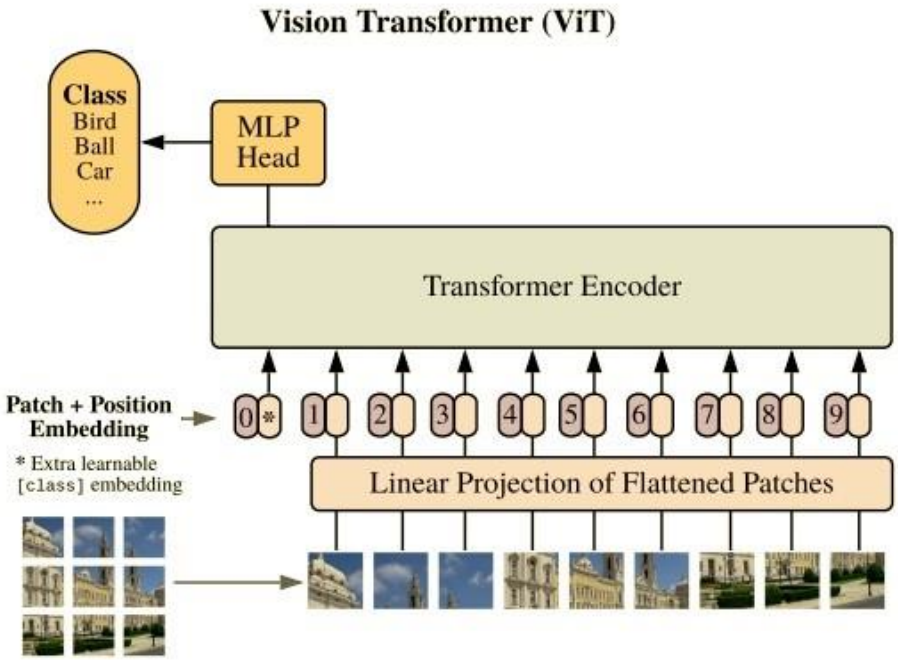


図 2.2 Vision Transformer の全体像
出典 [2]より引用

2.2.1　Transformer Encoder

次に Transformer Encoder の概要について図 2.2.1 に基づいて説明する．Transformer Encoder は，入力画像の特徴を抽出するためのものであり，Multi-Head Attention，Layer Normalization，MLP(Multi-Layer Perceptron)の 3 つで構成されている．Multi-Head Attention は入力画像の異なる情報の視点や側面を捉え，それらを統合してより豊かな表現を生成することで，モデルの性能を向上させる重要な役割をもつ．Layer Normalization は，各層の平均が 0，標準偏差が 1 になるように正規化する．それによって学習の安定性が向上し，特徴表現のスケールが均一化される．MLP(Multi-Layer Perceptron) は，Multi-HeadAttention によって生成された特徴を抽出する役割を担う．これにより，入力データの複雑な関係やパターンを捉えることができる．実際の処理の式は

$$Y=GELU(XW1+b1)W2+b2$$

となる．入力 X に行列 W1 をかけ，定数項 b1 を足し，活性化関数 GELU に通す．そしてもう一度行列 W2 をかけ定数項 b2 を足している．

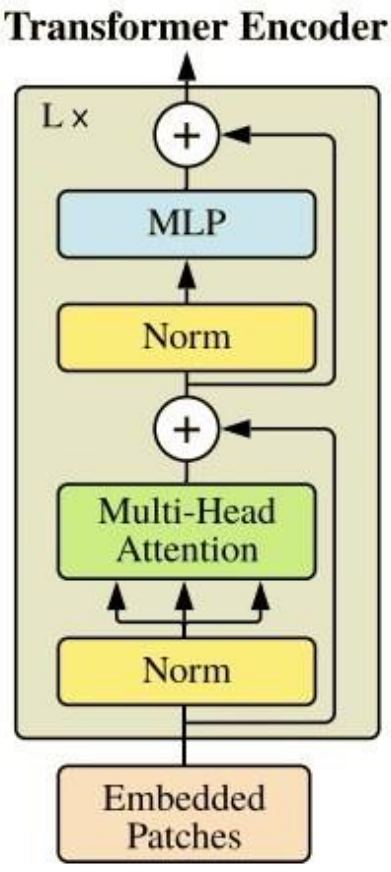


図 2.2.1　Transformer Encoder

出典　[2]より引用

2.2.2　Multi-Head Attention

Multi-HeadAttention の概要を図 2.2.2 に基づいて説明する．q(Q)が情報を問い合わせるためのキーとなる要素，k(K)が入力の各要素に関連付けられたキーベクトルで，q との類似性を計算するための要素，v(V)が重み付けされた情報を保持するための要素である．これらを用いて，Scaled Dot-Product Attention で画像のどこに注意するかを求める．それをヘッドの数だけ繰り返し，そのすべてのヘッドの結果を結合し，最後に重み行列を適用して出力を計算する．Linear 関数は，q,k,v の変換に使用される．

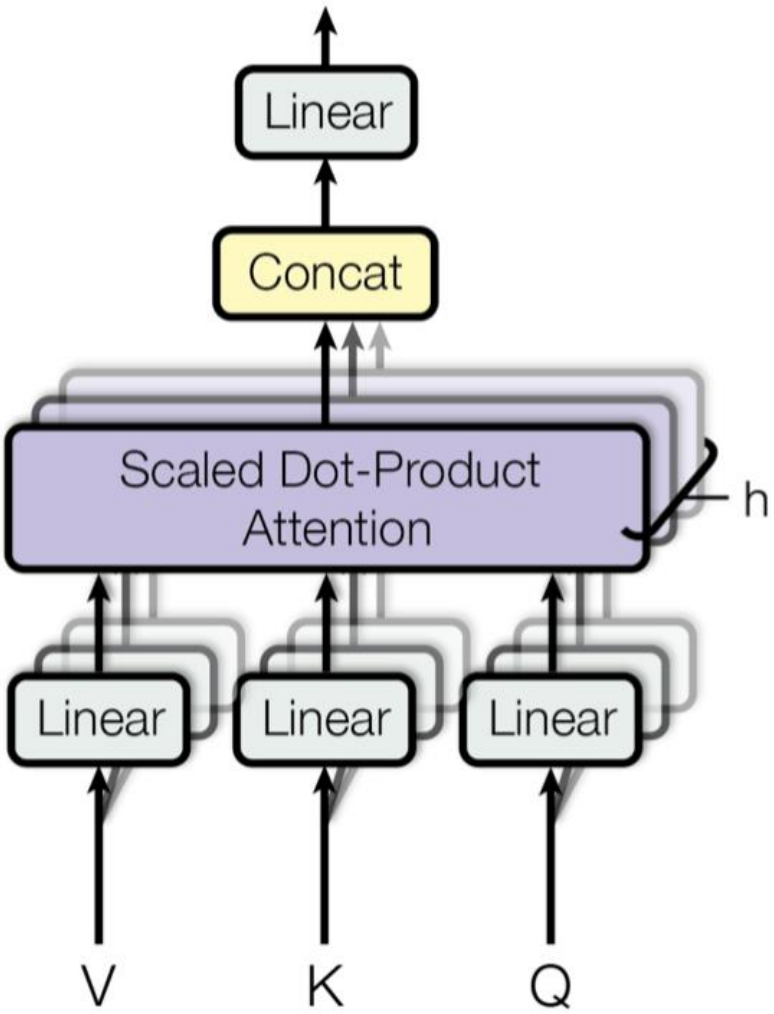


図 2.2.2　Multi-Head Attention

出典　[3]より引用

本研究で使用するモデルである vit_base_patch16_224 の処理の流れを図 2.2.3 に基づいて説明する．14*14 のパッチにクラストークンを追加した 197 個を入力として，正規化し重み行列をかけ，それぞれ q,k,v に変換する．続いて，Multi-Head Attention に入り，q に k の転置の行列積を求める．次に，softmax 関数に通して得られる attention matrix に対して v の行列積を求める．softmax 関数は，0 から 1 の範囲の値に変換し，それらの合計が 1 になる確率分布に変換する役割を持つ．この Multi-Head Attention の処理をヘッドの数である 12 回分繰り返し，それらを並べたものに fc にて重み行列をかける．最後に，正規化を行い，MLP を用いて特徴を抽出されるという流れになる．

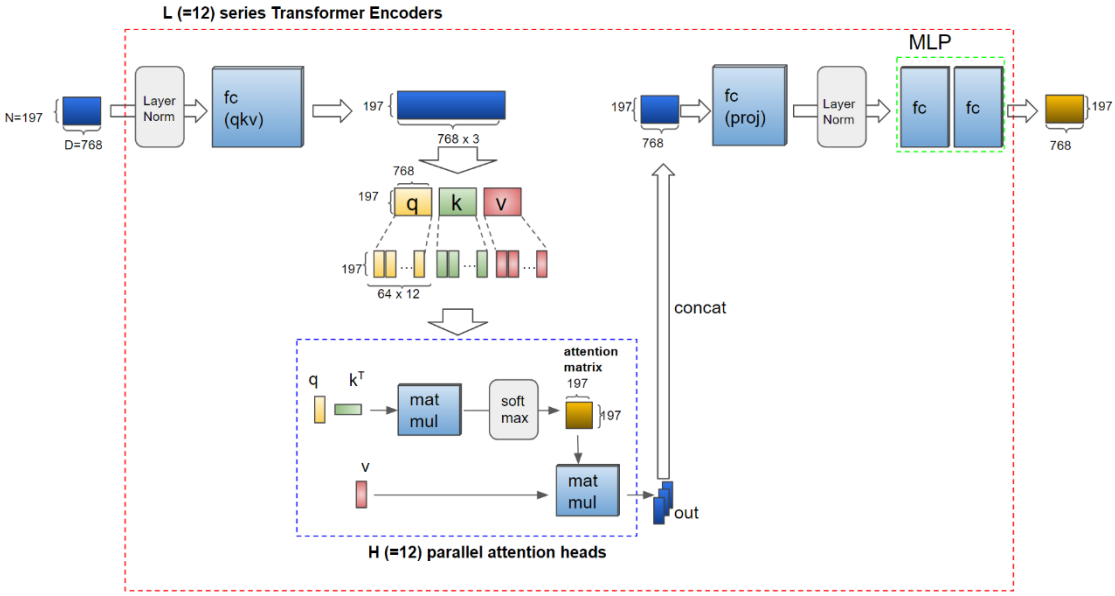


図 2.2.3 処理の流れを表す図

出典 [4]より引用

第 3 章　提案手法

3.1　学習環境

本研究では，Google Colaboratory を使用する．そして，学習においてバッチ数は 64，エポック数は 30 として研究を行う．

3.1.1　データローダの作成

犬と猫のデータセットをダウンロードし，データセットを作成する．その際，学習サンプルは，画像を 224*224 にリサイズし，ランダムに水平方向に反転・拡大して作成する．そのため，train_accuracy より test_accuracy のほうが評価が高くなる可能性があると考えている．

3.1.2　Multi-Head Attention での処理の可視化

本研究を行うにあたって，Vision Transformer の最大の特徴である Multi-Head Attention の実際の処理を可視化したいと考えている．Multi-Head Attention では，2.2 で説明した通り q,k,v を生成する際，より豊かな表現を得るために各ヘッドで別の重み行列が使われているはずであるのでそれを確認する．Attention Matrix を使い，各ヘッド毎の，画像のどこに注意しているかを可視化してヘッド 7 つ分出力させる．

3.2　使用するモデル

本研究で使用するモデルを挙げる．まず Vision Transformer では vit_base_patch16_224 を使用する．Transformer Encorder は 12 回繰り返し，Multi-Head Attention のヘッドの数は 12 となっている．

比較として用いる CNN のモデルは，ResNet の ResNet-152 と，EfficientNet の EfficientNetV2-S である．ResNet-152 は ResNet シリーズの中で最も大きなサイズのモデルの 1 つで，レイヤーの数が 152 ある．深いネットワーク構造を持ち，高い精度を提供する．EfficientNetV2-S は EfficientNet の中で最も小さいサイズのモデルで 44 個のブロックを持ち，精度と速度のバランスを重視した設計であり，リソースが制限された環境でも効果的な性能を発揮する．

尚，これらのモデルは事前学習されたモデルに対してファインチューニングを施して使用することもできる．

3.3　比較の手法

本研究では，比較の手法を 2 つ提案する．1 つ目は，事前学習の有無である．事前学習の有無によって評価が変わるのか，変わる場合はどれだけの差が出るのかを調査する．2 つ目は，学習サンプルの数を 4 パターンに分けて実験を行うということである．もともと Vision Transformer にサンプル数は数億枚必要とされているため，サンプル数を減らしていくことにより Vision Transformer の精度は他と比べてどうなるのかを調査するため，学習サンプルを 1000，100，50，20 と減らしていき検討を行う．

第 4 章　実験

4.1　Multi-Head Attention での処理

Multi-Head Attention での処理を，7 つ目のヘッドまで出力したものを図 4.1 に示す．この画像は，注目すべき箇所を明るい色で表しているものとなっている．7 つ全て注目すべき箇所が変わっているため，それぞれに異なる重み行列がかけられていることがわかる．

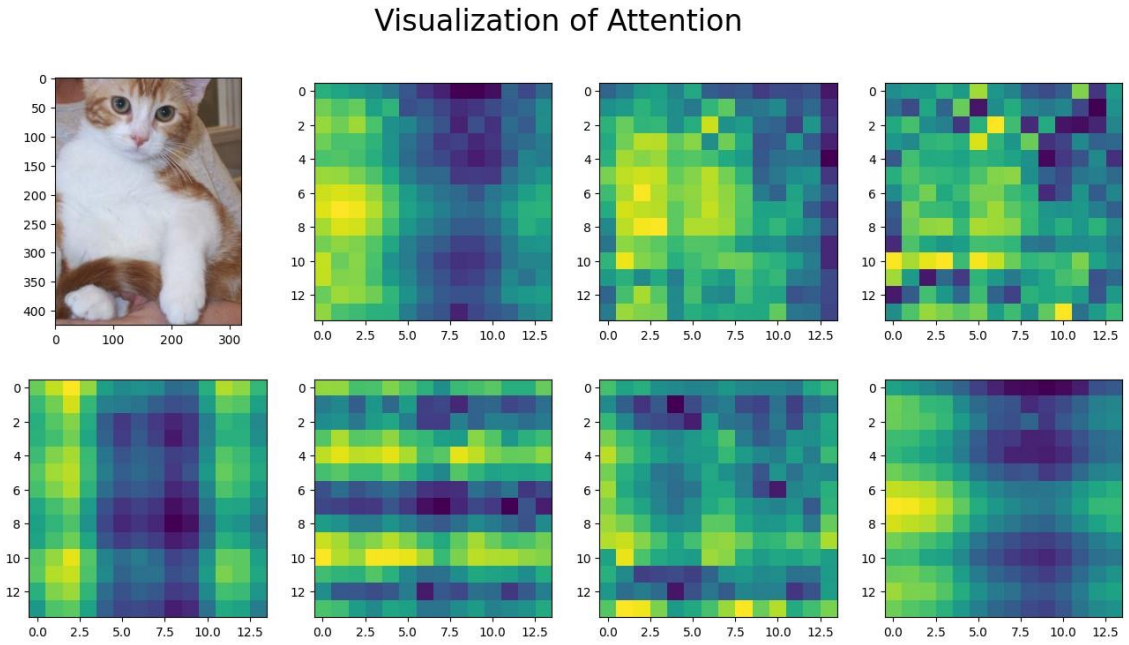
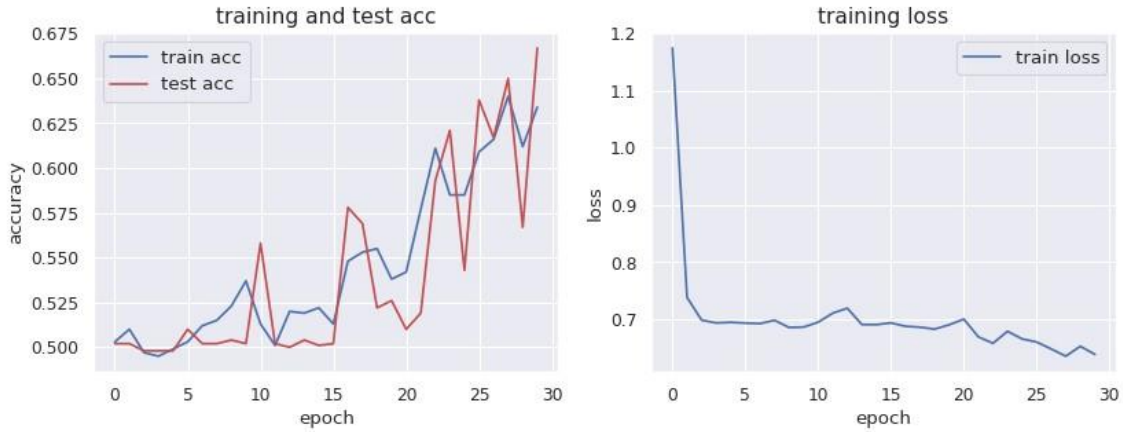


図 4.1　ヘッド毎の注目度

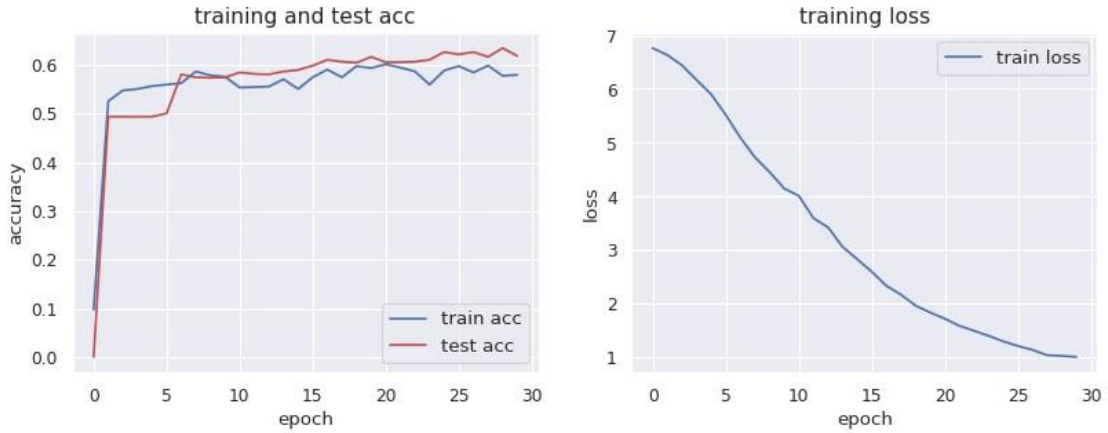
4.2　事前学習無し

次に, 犬と猫の判別を, 事前学習を無しにして学習させた. Vision Transformer, ResNet,, EfficientNet の 3 つの比較を，学習サンプル数が 1000 の場合を図 4.2.1(a)及び(b)及び(c)に，学習サンプル数が 100 の場合を図 4.2.2(a)及び(b)及び(c)に，学習サンプル数が 50 の場合を図 4.2.3(a)及び(b)及び(c)に，学習サンプル数が 20 の場合を図 4.2.4(a)及び(b)及び(c)に示す.



(a)Vision Transformer

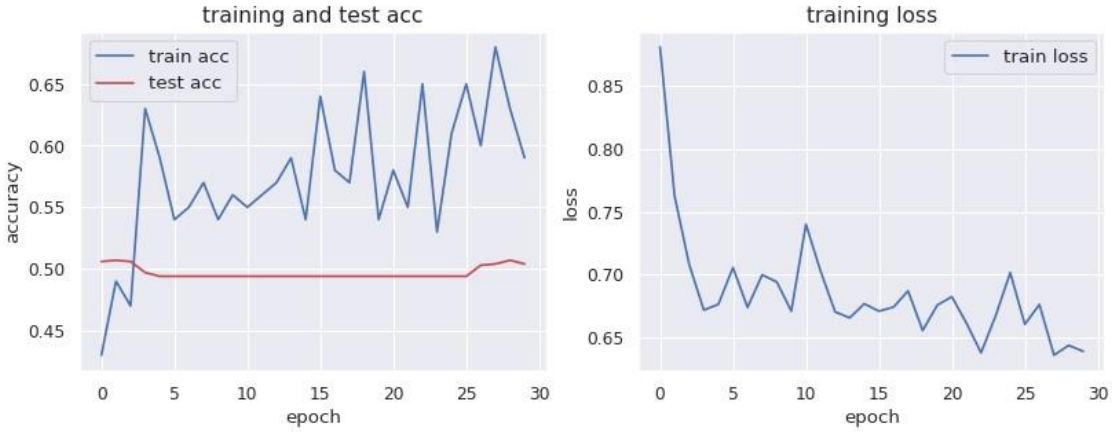
(b) ResNet



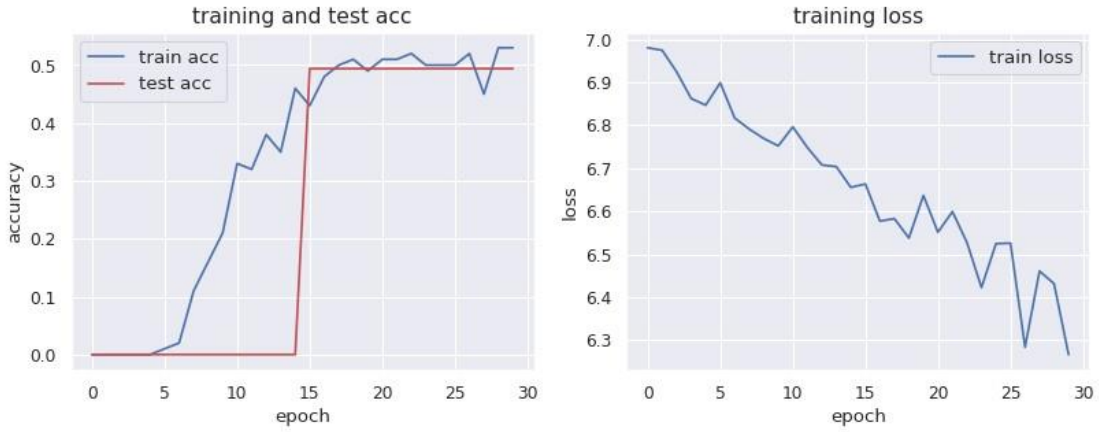
(c)EfficientNet

図 4.2.1 学習サンプル数が1000の場合

(a)Vision Transformer

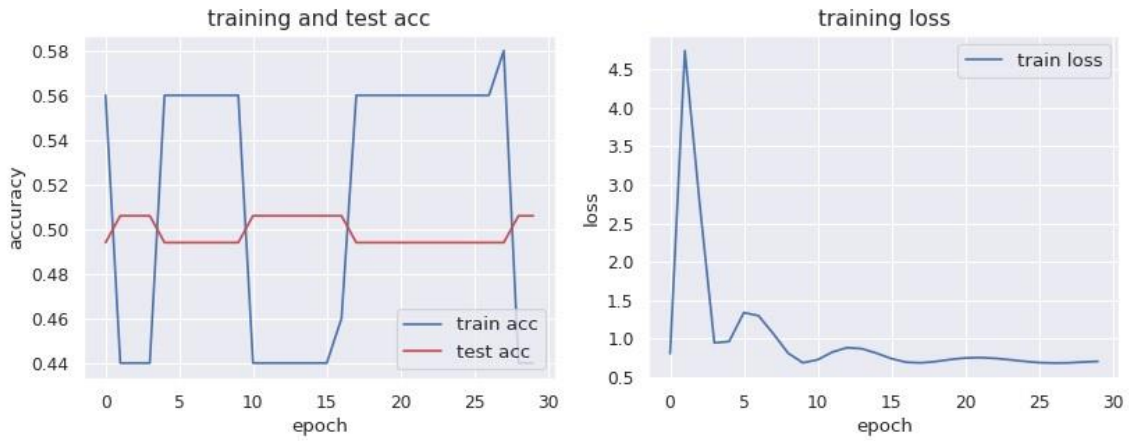


(b)ResNet



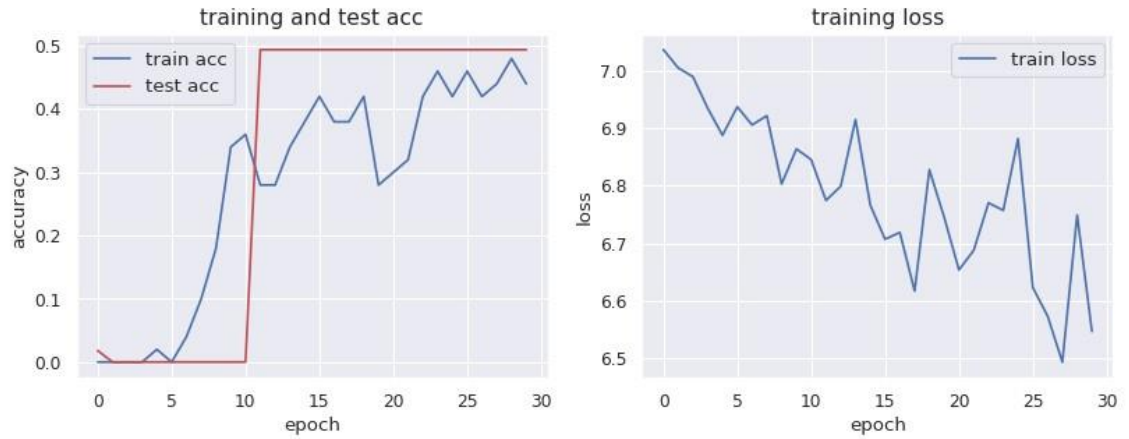
(c)EfficientNet

図 4.2.2 学習サンプル数が100の場合



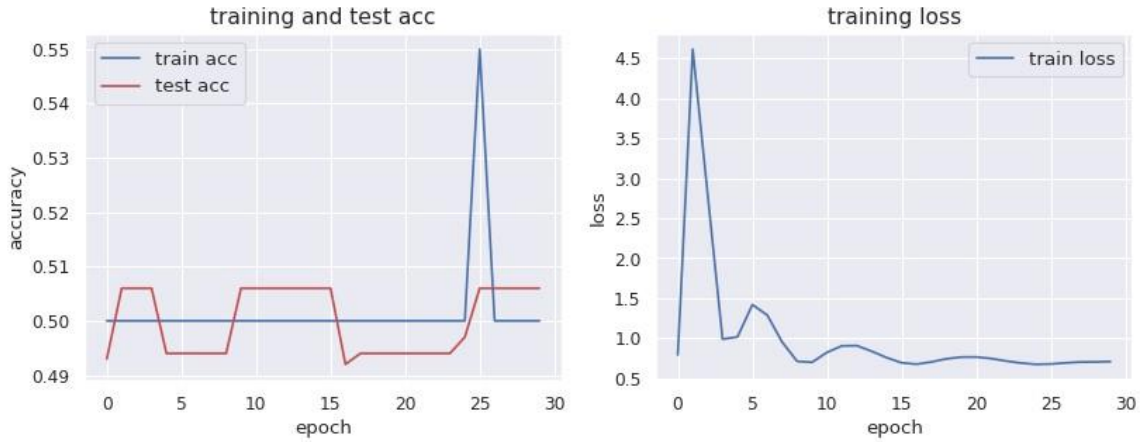
(a)Vision Transformer

(b)ResNet

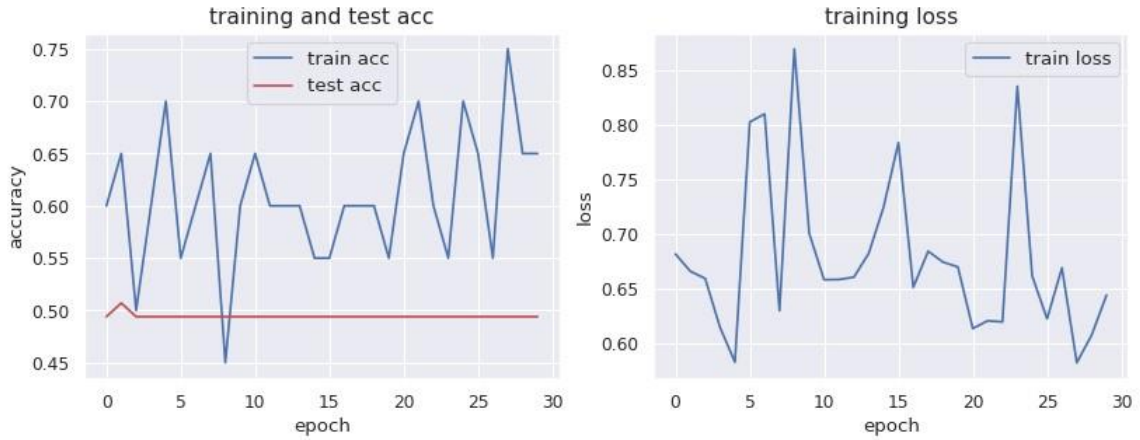


(c)EfficientNet

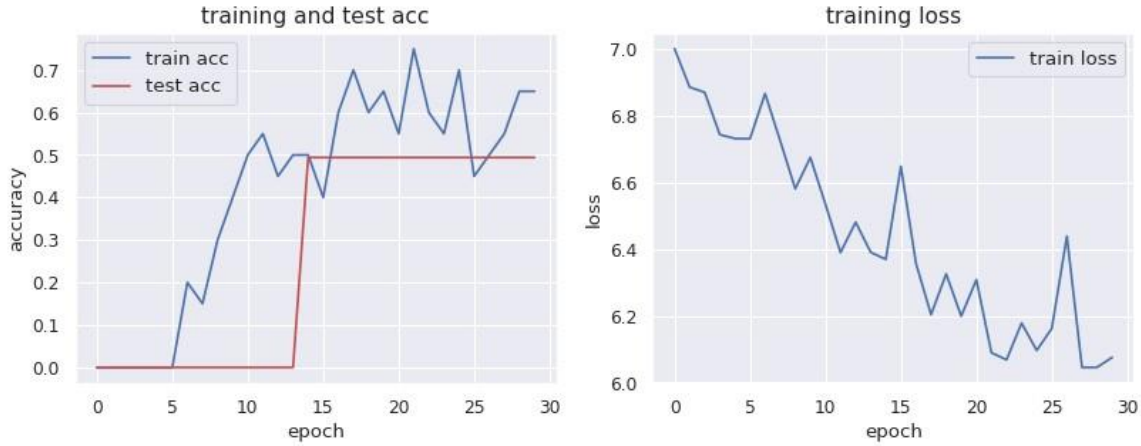
図 4.2.3 学習サンプル数が50の場合



(a)Vision Transformer



(b)ResNet

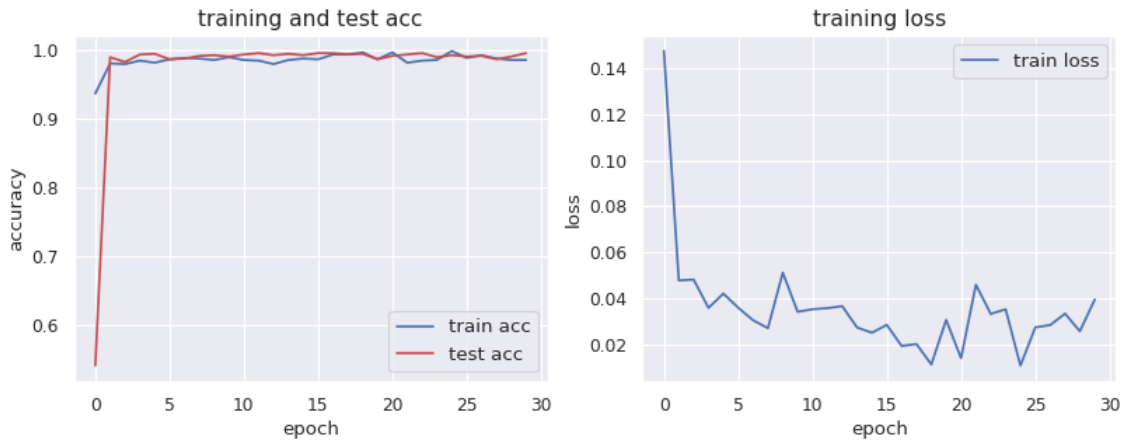


(c)EfficientNet

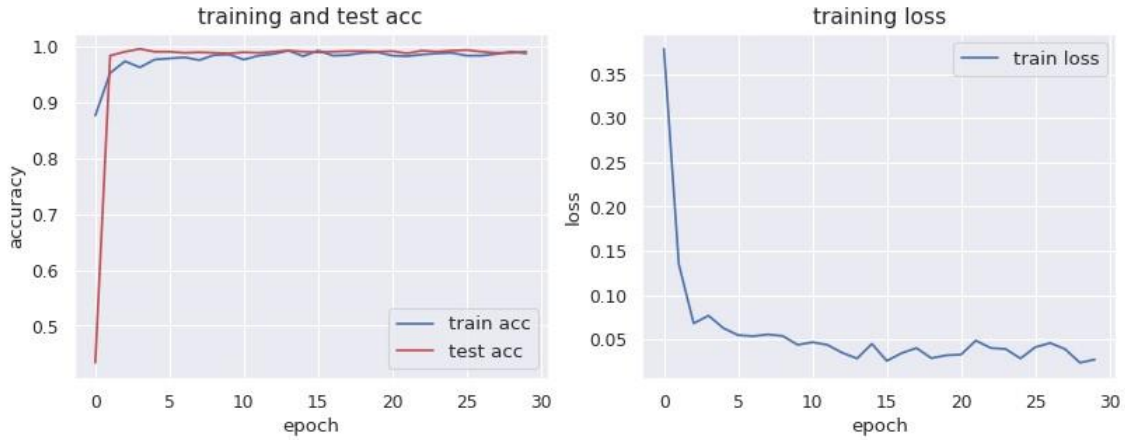
図 4.2.4 学習サンプル数が20

4.3　事前学習有り

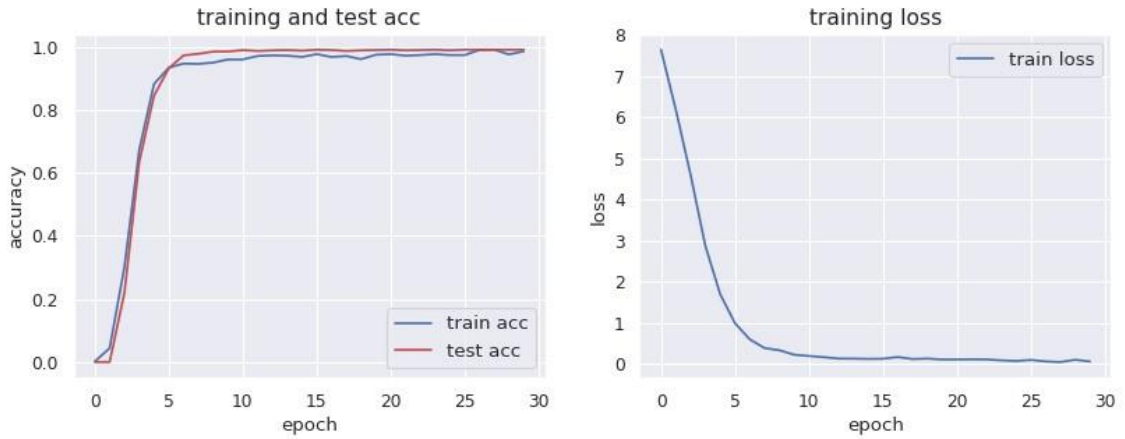
次に, 犬と猫の判別を, 事前学習を有りにして学習させた. Vision Transformer, ResNet,, EfficientNet の 3 つの比較を，学習サンプル数が 1000 の場合を図 4.3.1(a)及び(b)及び(c)に，学習サンプル数が 100 の場合を図 4.3.2(a)及び(b)及び(c)に，学習サンプル数が 50 の場合を図 4.3.3(a)及び(b)及び(c)に，学習サンプル数が 20 の場合を図 4.3.4(a)及び(b)及び(c)に示す.



(a) Vision Transformer

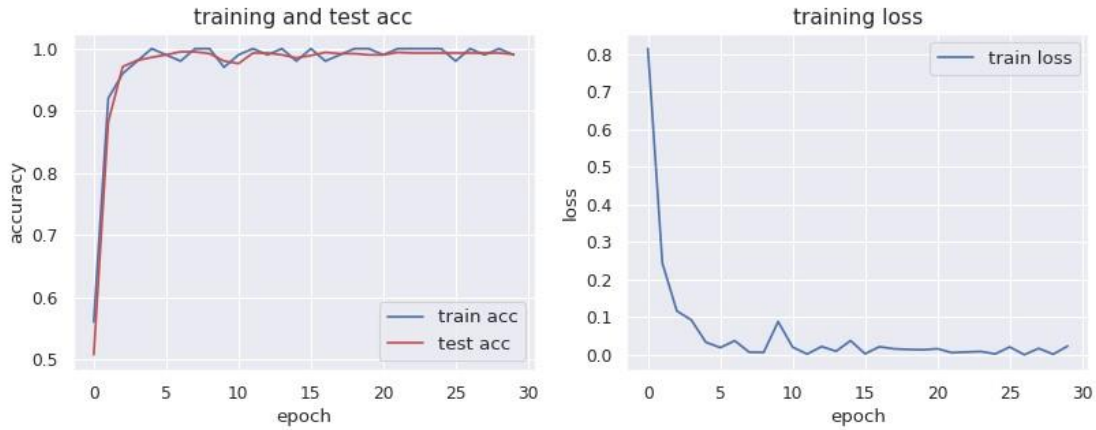


(b) ResNet

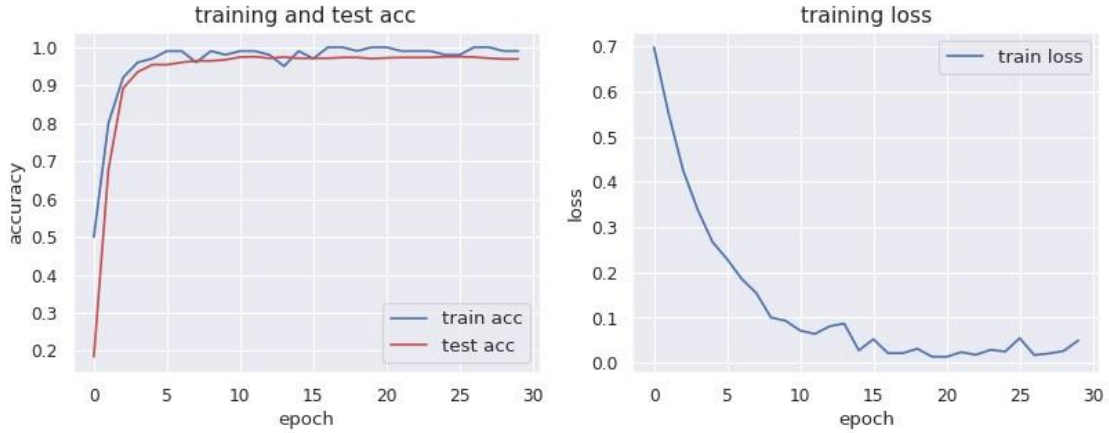


(c)EfficientNet

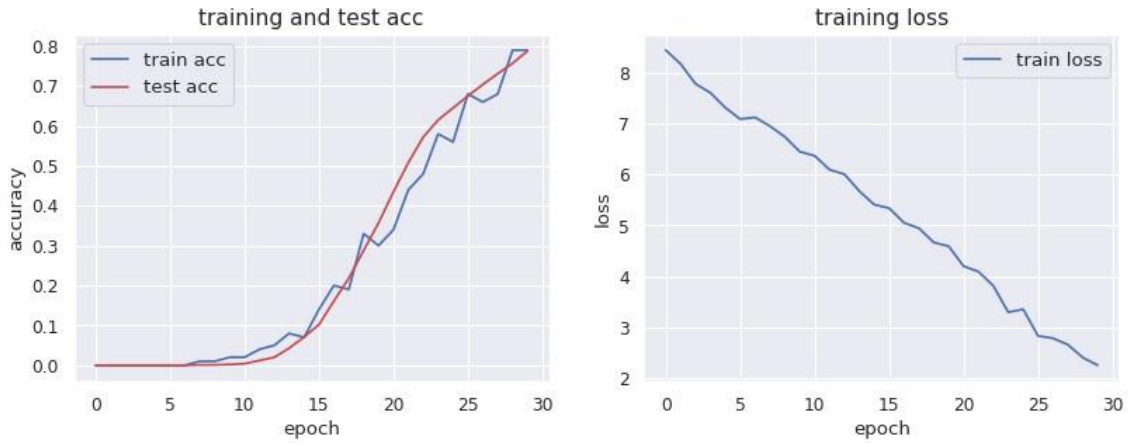
図 4.3.1 学習サンプル数が 1000 の場合



(a)Vision Transformer

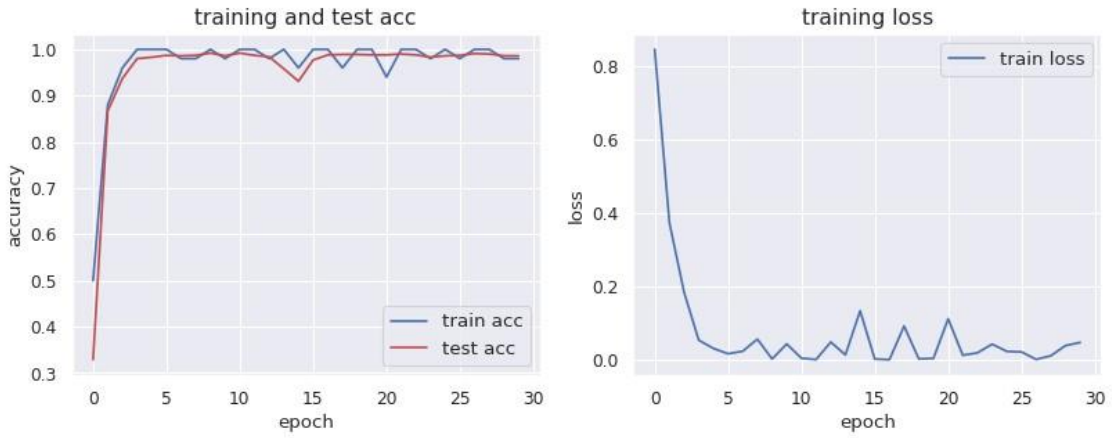


(b)ResNet

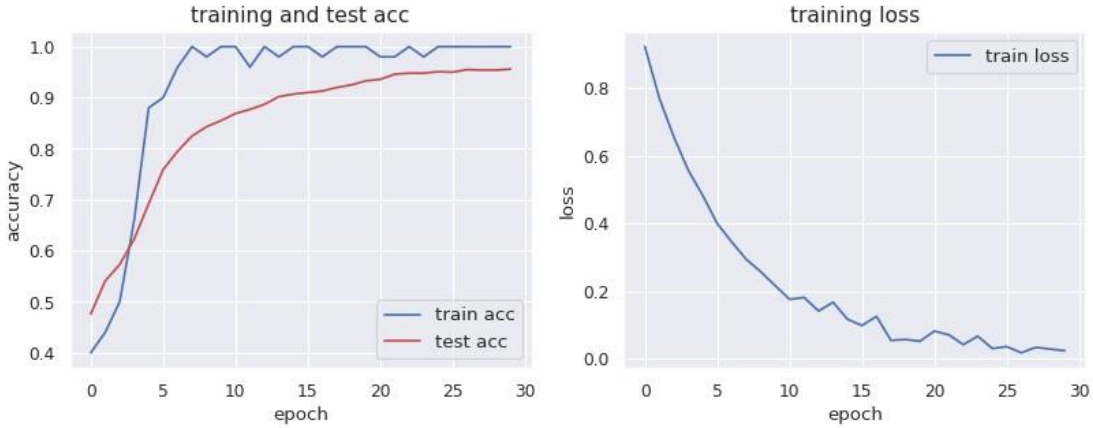


(c)EfficientNet

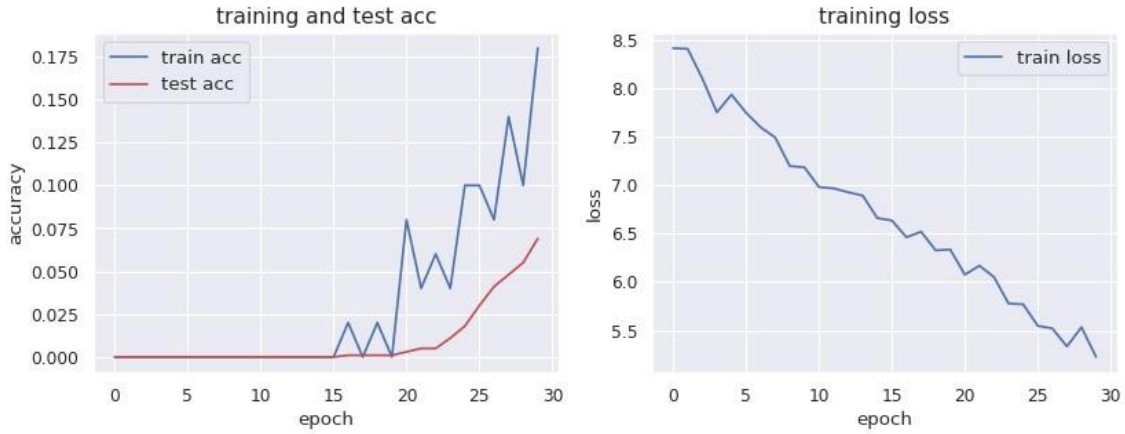
図 4.3.2 学習サンプル数が100の場合



(a)Vision Transformer

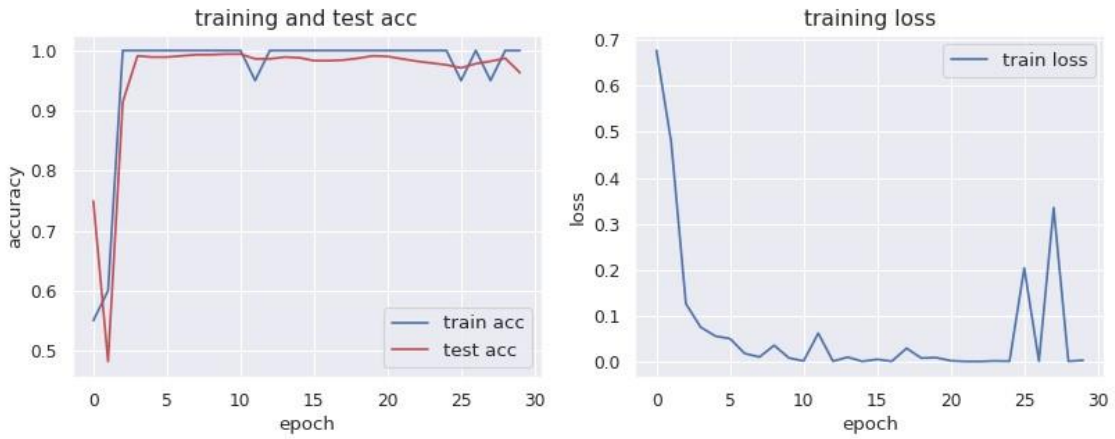


(b)ResNet

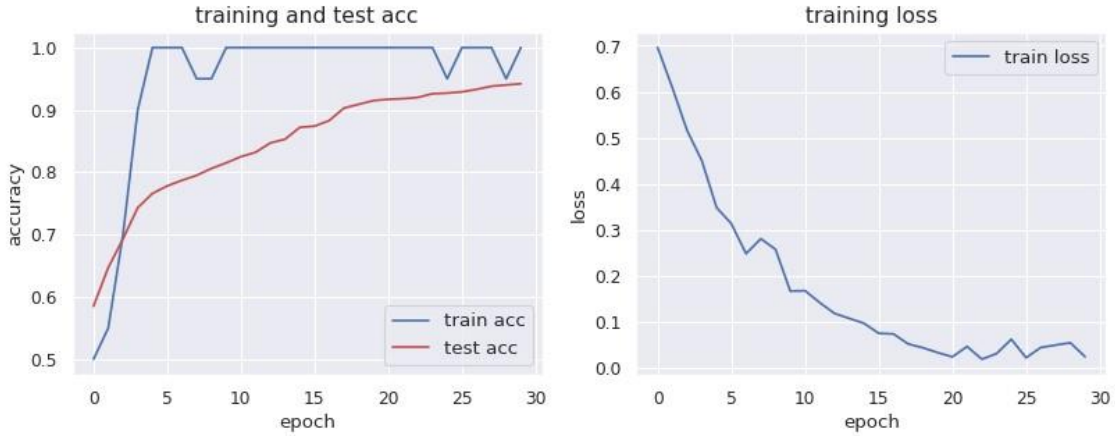


(c)EfficientNet

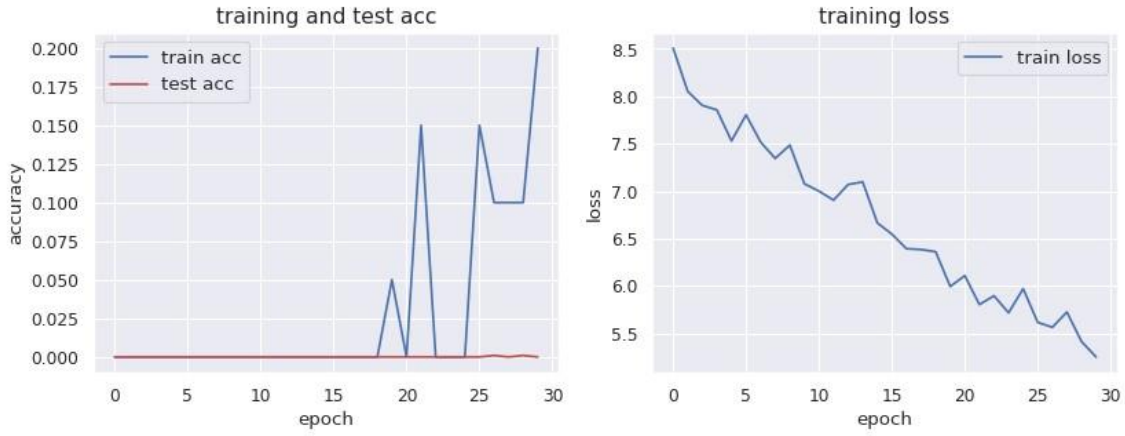
図 4.3.3 学習サンプル数が50の場合



(a)Vision Transformer



(b) ResNet



(c)EfficientNet

図 4.3.4 学習サンプル数が20の場合

4.4 考察

まず最初に，事前学習無しの Vision Transformer に着目する．学習サンプル数が 1000 の場合（以後，学習サンプル 1000）の test_accuracy の値は，学習を重ねていく中で大きなブレがあり，最高でも 66.7%という結果になった．また，他の学習サンプル 100, 50, 20 はどれも test_accuracy が 50%前後付近から上がることはなかった．ResNet や EfficientNet においても，ほぼ同様の結果となったことが読み取れ，どのモデルも 70%を超えることがなかった．よって，学習サンプル 1000 では精度がかなり悪く，学習サンプル 100 からはモデルがほぼ機能しないということが分かった．この結果を踏まえ，事前学習有りの Vision Transformer に着目すると，学習サンプル 1000 の test_accuracy の値は，1 エポック目から 99.0%，そこから先は 99.5%前後の出力となっている．そして，学習サンプルが 100 の場合は 99.2%前後，学習サンプルが 50 の場合は 99%前後，学習サンプルが 20 の場合は 97%前後であることがわかる．この時点で，事前学習有りのほうが優れていることがわかる．

次に，Vision Transformer，ResNet，EfficientNet の 3 つの精度の違いに着目する．その際，事前学習無しより有りのほうが精度が良いということがわかっているため，事前学習有りのほうで比較をする．最後の 5 エポックの出力の平均で比較すると，学習サンプル 1000 の場合では Vision Transformer が 99.62%，ResNet が 98.98%，EfficientNet が 99.10%となり，学習サンプル 100 の場合では，Vision Transformer が 99.26%，ResNet が 97.16%，EfficientNet が 73.10%となった．さらに，学習サンプル 50 の場合では，Vision Transformer が 98.80%，ResNet が 95.38%，EfficientNet が 4.86%となり，学習サンプル 20 の場合では，Vision Transformer が 97.62%，ResNet が 93.64%，EfficientNet が 0.04%となった．Vision Transformer の精度がほとんど落ちていないことの理由として，それぞれのパッチの位置情報を保持していることで，局所的な特徴だけでなく大きな特徴や局所的な特徴の組み合わせまで得られたことにより，性能の向上につながったからだと考えた．

第 5 章　結論

本研究では，学習サンプルの数が少数の状況下での Vision Transformer の認識性能について議論した．ここでは，認識課題として犬と猫の認識問題を扱った．同様の実験を ResNet および EfficientNet を用いて行い，Vision Transformer との精度を比較した．その際，事前学習を有りにした場合と無しにした場合に分けて比較した．

事前学習無しの方では，どのモデルの test_accuracy も 70%を超えることがなく，振るわない結果となった．事前学習有りの方では，学習サンプル数として 1000，100，50，20 を用いた全ての場合において，ResNet 及び EfficientNet より Vision Transformer の精度が優れていた．よって，学習サンプル数が少ない状況下で事前学習を用いれば Vision Transformer は最も有望であることが分かった．

本研究では，最大サンプル数が 1000 と少なく，エポック数が 30 という条件下で行ったため，すべての条件下で断言できるわけではないが，少なくともこの条件下では，画像の学習において，事前学習をしないということは，ほとんど機能しなくなるほど精度が悪くなるということが言える．逆に，事前学習有りだとサンプル数が少数でも高い精度を保ってくれるということが言える．

参考文献

[1] Learning Blog <体験型>学習ブログ「畳み込みニューラルネットワーク（CNN）の基本形」
URL: https://zero2one.jp/ai-word/basic-form-of-cnn/
[2] Qiita「【PyTorch】Vision Transformer（ViT）を自作しながら解説する」
URL: https://qiita.com/zisui-sukitarou/items/d990a9630ff2c7f4abf2
[3] Multi-Head Attention
URL: https://paperswithcode.com/method/multi-head-attention
[4] Vision_Transformer_Tutorial.ipynb「Unofficial Walkthrough of Vision Transformer」
URL:https://colab.research.google.com/github/hirotomusiker/schwert_colab_data_storage/blob/master/notebook/Vision_Transformer_Tutorial.ipynb
[5] DeepSquare Media「画像認識の革新モデル！脱 CNN を果たした Vision Transformer（ViT）を徹底解説！」
URL: https://deepsquare.jp/2020/10/vision-transformer/