第2章　方法

## 2.1 畳み込みニューラルネットワーク

本検証では, 訓練済み畳み込みニューラルネットワークをfine-tuningし, 物体判別タスクの検証を行った.

### 2.1.1 二次元畳み込みニューラルネットワーク

二次元の畳み込みニューラルネットワークで画像中の物体判別タスクを行うネットワークとして,ImageNet(引用)を用いた1000クラス判別タスクでpre-trainされた ResNets (引用) を用いた. 本検証においては, 50層のResNetsを用いた.

### 2.1.2 三次元畳み込みニューラルネットワーク

本研究では三次元の畳み込みニューラルネットワークとして, 画像中の物体判別タスクでpre-trainされた二次元の畳み込みニューラルネットワークを拡張した時空間畳み込みニューラルネットワークと, 同様のネットワークを動画中の動詞判別でpre-trainしたネットワークを用いた.

#### 2.1.2.1 畳み込みニューラルネットワークの拡張

時空間方向の畳み込みを行う三次元の畳み込みニューラルネットワークで, 2次元画像中の物体判別タスクを行うものとして, I3D (引用) ネットワークを用いた.

I3Dネットワークは, 訓練済みの二次元畳み込みニューラルネットワークを3次元に拡張することにより作られる時空間畳み込みニューラルネットワークである. 拡張は, 時空間畳み込みニューラルネットワークの作成と, 2次元畳み込みニューラルネットワークからの学習済みの重みの転移によって行われる. 時空間畳み込みニューラルネットワークは, 畳み込みニューラルネットワークの畳み込み層とプーリング層に時間方向の次元を加えることにより作成される. ネットワークを作成した後の重みの転移は, 時空間畳み込みニューラルネットワークに二次元の同じ画像を繰り替し, 作成された動きがない動画 (boring-video) を入力した時の出力が, もとの二次元畳み込みニューラルネットワークに同じ画像を入力した時の出力と等しくなるような制限をみたすように行う.

本検証では二つの方法で, 上記の制約を満たす拡張を行った. それぞれの方法において, 時空間畳み込みニューラルネットワークの畳み込み層の重みは, 変換前の2次元畳み込みニューラルネットワークにおいて対応する畳み込み層の重みから転移を行った. 一つ目は, 変換する層の時間軸方向の大きさがNのとき, 対応する畳み込み層の重みを時間方向にN回繰り返した後に, 1/N倍することにより時空間ネットワークの重みの初期化を行った. 二つ目の方法では, 時空間畳み込みニューラルネットワークの重みをすべて0で初期化を行った後に, 時間軸において中央に位置するフィルターのみを対応にする二次元畳み込みニューラルネットワークの重みを用いて初期化することによって転移を行った. 本研究においては, 前者を平均化拡張, 後者を中心化拡張と呼ぶ.

本検証においては, 画像中の物体判別にpre-trainされた時空間畳み込みニューラルネットワークとして, ImageNetでpre-trainされたResNets50 をそれぞれ, 平均化拡張, 中心化拡張によって時空間畳み込みニューラルネットワークに拡張したものを用いた.

#### 2.1.2.2 動詞判別時空間畳み込みニューラルネットワーク

## 2.2 データセット

### 2.2.1 Moments In Time データセット

I3Dの訓練, および検証にはMomets In Timeデータセット (引用) から抽出した1250件の動画データ及び, 動画に対応する物体カテゴリラベルを使用した. Moments In Timeデータセットは100万枚以上の3秒間の動画に339種類のアクションのカテゴリが動詞名で一つずつ付けられたデータセットであり, 同様のものとしては最大規模のデータセットである.

### 2.2.2 データセットの抽出

本研究では, 動詞ラベルではなく動画中の物体カテゴリラベルを利用するため, Moments In Timeデータセットから訓練, バリデーション用のデータとして1200件, テスト用に50件のデータを抽出した. 抽出に際して訓練, バリデーション用データは150のアクションカテゴリから8件ずつ, テスト用データはその内50のカテゴリから1件ずつのデータを抽出した.

抽出した動画に対する物体カテゴリのラベリングは, それぞれの動画中に確認できる物体のラベルを複数つける形で行った.

ラベリングを行った結果, 193の物体カテゴリがラベルとして与えられ, 1動画あたりの平均ラベル数は1.41であった. 本研究では, この内出現頻度上位20ラベルのみを用いた. 上位20のラベルがつけられた動画は937件, 1動画あたりの平均のラベル数は1.25であった.

ラベルが付けられた動画データは, 全て時間が3秒間, フレーム数90枚, 解像度は縦256画素, 横256画素のサイズの物であった. 本検証においては, 90フレームから均等に32フレームを抽出して用いた.

## 2.3 物体判別学習

### 2.3.1二次元畳み込みニューラルネットワーク

二次元の畳み込みニューラルネットワークは以下の方法で訓練を行った。ニューラルネットワークへの入力は、作成したデータセット中の動画データの32フレームをそれぞれ一枚の画像として入力を行った。入力は全動画の全フレームをランダムにシャッフルした後、16枚を1バッチとして入力を行った。また、それぞれの入力画像に対して、左右、上下の反転をおこなった後、256 x 256の解像度の画像から224 x 224の解像度の画像をランダムな位置で切り抜く前処理を行った。

また、学習時の条件は以下のものを用いた。損失関数には最終層の出力にシグモイド関数を適用した各ラベルの予測値と, 真のラベルとのクロスエントロピーの全ラベル間での平均を用いた. 最適化手法としては, Momentum SGD (引用?) を, Momentumの値を0.9として使用した. 重みは4バッチ毎に勾配を蓄積し、その勾配を用いて更新した。本研究においては, 一回の重みの更新を1ステップと呼ぶ。学習率は初期値として0.01を用い, それぞれ300ステップ, 1000ステップの学習後に0.1倍した. また, 学習の際はWeight Decay(引用？) を用いた正則化を行った.

2.3.2 時空間畳み込みニューラルネットワーク

二次元の畳み込みニューラルネットワークは以下の方法で訓練を行った。ニューラルネットワークへの入力は、1動画を1バッチとして入力を行った。また、それぞれの入力動画に対して、左右、上下の反転をおこなった後、256 x 256の解像度の動画から224 x 224の解像度の動画をランダムな位置で切り抜く前処理を行った。

また、学習時の条件は、上述の二次元畳み込みニューラルネットワークと同様のものを用いた。

2.3 検証