DEIM Forum 2021 G21-5

# モバイルアプリケーションにおけるUIデザイン自動評価の検討

栗林　峻†　酒井　哲也†

† 早稲田大学基幹理工学研究科情報理工・情報通信専攻　〒 169 – 8555 東京都新宿区大久保 3-4-1

E-mail: †kuri884@toki.waseda.jp, ††tetsuyasakai@acm.org

**あらまし**　ユーザインターフェース (以下，UI) デザインに関する研究は数多く行われてきているが，デザイン評価は，実際に人手で画面を操作してデータ収集を行ったり，使用者からのフィードバックをもとに分析したりという手法が主である．これらは人手と時間かかることや，定量的な分析が難しい点が課題であった．本研究では，UI デザイナーであるユーザを想定し，大規模な UI データセットである Rico を利用して CNN を学習させ，UI 画像からリスト形式の画面を分類する分類モデルと，ユーザビリティを評価する評価モデルとを生成し，それらを用いたフィードバックを検討した．分類モデルを生成する過程において，ImageaNet により事前学習したモデルをファインチューニングする手法が効果的であることがわかった．また，分類モデルが予測において重要視している点を Grad-CAM を用いて可視化して解釈した．

**キーワード**　ユーザインターフェース，モバイルアプリ，CNN，デザイン評価

## 1 はじめに

ひとことに UI といっても，そのデザインには求められ，提供される機能によって様々なものが存在する．UI は人間の細かな感覚に近い部分の技術であるからこそ，自動化や定量的な評価が難しい分野であった．過去にも UI デザインに関する研究は数多く行われているが，UI デザインの部分においては，経験則的であったり，個別具体的な作業が多く，知見が定量化されてこなかった．また評価に関しては，実際に人手で操作してデータ収集を行ったり，使用者からのフィードバックをもとに分析したりといった手法が主であり，時間と手間がかかることから，定量的な評価が難しい点が課題であった．

ただし，近年では機械学習手法の発展により，UI に関する分野でも自動化に関する研究が活発に行われるようになってきている．また，昨今での普及により，PC・スマートフォンの利用者はこれまで以上に幅広いものとなっており，中でもスマートフォン等のモバイル機器は，最も身近な端末として多くの場面で利用されている．スマートフォンは，App Store や Google Play といったアプリストアで提供されるアプリをインストールすることで，必要な機能に素早くアクセスすることが可能となり，ユーザビリティを向上させている．アプリと，それを操作する人間とを仲介するのが UI であるが，利用者の増加，層の拡大，提供される機能の多様化により，使いやすい UI デザインの価値は一層高まっている．

本研究では，UI デザイナーであるユーザの補助を想定した UI デザインの自動評価を目標として，UI に関する大規模なデータセットである Rico を利用し，ファインチューニング用いて CNN を学習させ，UI デザイン分類モデルと，UI デザイン評価モデルとを生成した．そして，これらのモデルを用いて，UI デザインをシステムが自動で評価するする手法を提案した．また，モデルが予測を行うにあたって判断材料とした部分の解釈への検討を行った．

## 2 関連研究

本節では，本研究に関連する研究について説明をする．

### 2.1 マテリアルデザイン

Google 社 [1] は，UI デザインのガイドラインとして，画像に影を示すことで奥行きを示すことや，色彩による強調などの項目を盛り込んだ「マテリアルデザイン」[1]を提案し，これを公開している．また，このガイドラインを元に UI を実装するためのライブラリ [2]を提供している．

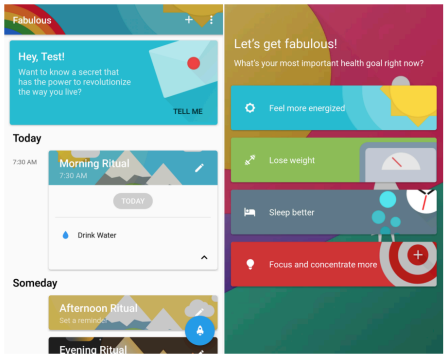


図 1　マテリアルデザイン（文献 [1] より引用）

### 2.2 UI デザインの定量的評価

Zexun ら [2] は，これまでの UI 設計と評価は，デザイナーや専門家による経験則的な作業に偏っているとして，増加するアプリの数に対応できない点，定量的な分析が難しい点を課題として挙げている．また，UI デザインに関する論文を収集し，言

1：https://material.io/design

2：https://material.io/develop

及される頻度が高かった “Consistency”, “Hierarchy”, “Contrast”, “Balance”, “Harmony” について分析し，これまで定性的に議論されていた各項目について，定量的なガイドラインの提案を行っている．Zexun らが示したガイドラインの例を図 2 に示す．

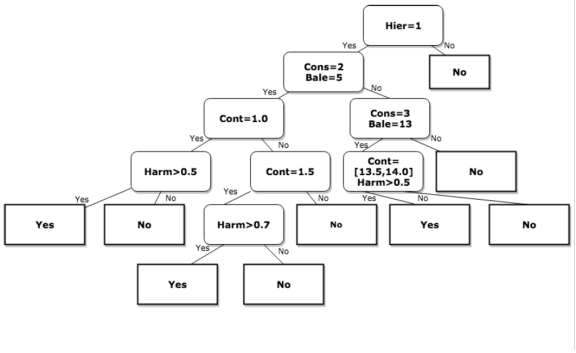


図 2　Zexun らによる UI 作成ガイドライン（文献 [2] より引用）

### 2.3　UI に関するデータセット

本研究では，Deka ら [3] による Rico dataset を用いる．UI に関する大規模なデータセットには，Rico の他に同じく Deka ら [4] による ERICA などが挙げられる．共に，モバイルアプリの UI スクリーンショットや，ユーザの操作に対するアプリケーションの応答等のデータを提供しているが，Rico はこれらのデータ数において ERICA の 4 倍近い規模である．Rico dataset についての詳細は次節にて述べる．

### 2.4　Rico の活用事例

#### 2.4.1　UI デザインを探索タスク

Huang ら [5] は，本研究でも用いる Rico を利用し，UI デザインの手書きのスケッチを元に，データセット内の類似した UI デザインを探索し，それを提示することで，デザイン設計の効率を高めるシステムを提案している．図 3 に，Huang らが提案するシステムの出力例を示す．

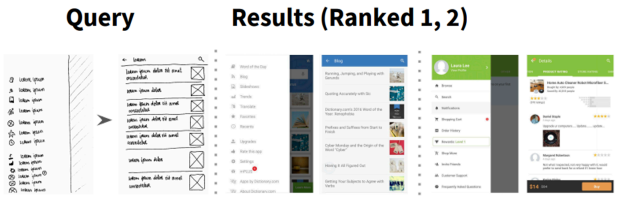


図 3　Huang らによるスケッチシステム（文献 [5] より引用）

#### 2.4.2　タップ可能要素認識分類タスク

Swearngin ら [6] は，モバイルアプリにおける UI 上の要素について，ユーザがタップ可能と認識するか否かを自動評価するシステムを提案している．この過程において，大規模な UI デザインのデータセットとして Rico を活用している．図 4 に，Swearngin らが提案するシステムの出力例を示す．

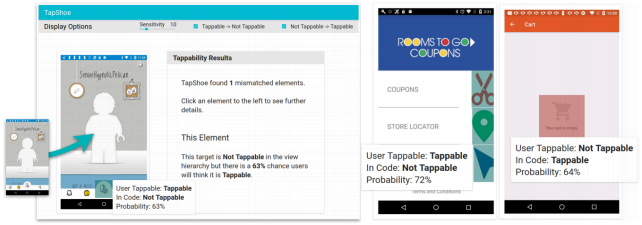


図 4　Swearngin らによるタップ視認性の研究（文献 [6] より引用）

### 2.5　ファインチューニング

ファインチューニングは，畳み込みニューラルネットワーク（CNN）において，すでに学習が済んでいるモデルの各層における重みを初期値として設定し，生成するモデルの学習を始めることで，学習を効率的に行う手法である．1,400 万枚以上の自然画像のデータベースである ImageNet[3]での 1,000 クラス分類問題を学習したモデルが公開されており，自然画像の分類問題において，利用されている．

### 2.6　Grad-CAM

Selvaraju ら [7] により提案された Grad-CAM（勾配加重クラス活性化マッピング）は，CNN による分類において，最後の畳み込み層の各特徴量マップにおける寄与度を用い，学習モデルが予測を行う上で重要とみなしている箇所をヒートマップとして出力し，可視化する手法である．図 5 に示すのは，モデルが画像に猫が写っていると予測した時の出力のヒートマップを Selvaraju らが示した例である．

(a) Original Image　(b) Guided Backprop ‘Cat’　(c) Grad-CAM ‘Cat’

図 5　Selvaraju らによるヒートマップの例（文献 [7] より引用）

## 3　データセット

本研究で用いる Rico について述べる．

Rico は Deka ら [3] により，モバイルアプリケーションに関する，デザイン検索，UI レイアウト生成，UI コード生成，ユーザーインタラクションモデリング，ユーザー知覚予測の研究のために生成された大規模なデータセットである．Google Play Store 上で提供されている 9,772 個のアプリの情報を人手とクローラにより収集し，それを提供している．Liu ら [8] はさらに画面上の要素を詳細に検出する研究を行っている．本研究で用いた項目を以下に挙げる．

3：http://www.image-net.org/

### 3.1 UI スクリーンショット

人手とクローラによりアプリケーションを操作し，66,000 件以上の UI スクリーンショットを提供している．

### 3.2 レイアウトベクトル

Rico では，UI スクリーンショット内の要素をテキスト部分と画像部分を区別して抽出し，その位置と形状に基づいたレイアウト情報をもとにオートエンコーダを学習し，64 次元のベクトル表現に圧縮して提供している．このベクトル表現で近傍探索を行うことで，データセット内の異なるアプリケーショから，類似したレイアウトの UI スクリーンショットの検索を可能にしている．図 6 に，Deka らが示した近傍探索の結果を示す．

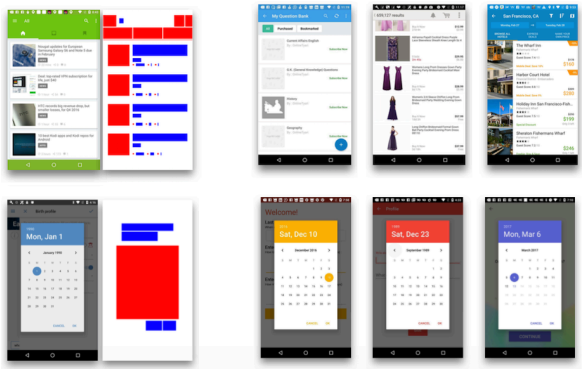

図 6　Deka らによる Rico での近傍探索の例（文献 [3] より引用）

### 3.3 アプリケーションのメタデータ

“ショッピング” や “教育” といった，Play Store でのアプリのカテゴリや，アプリのダウンロード数，ユーザによるレーティング，レーティングの投稿数，コメント数などのメタデータである．

レーティングはユーザがアプリを 0〜5 点の 5 段階で評価するもので，数値が大きいほどアプリの評価が高いことを示す．0.1 刻みの 5 点満点で，データセット内の全アプリケーションの平均レーティングは 4.1 である．

## 4 提案手法

本研究では，UI デザイナーであるユーザを想定し，ユーザにより入力された UI 画像を，リスト型（後述）か not リスト型かに分類し，リスト型に分類された UI 画像のユーザビリティを評価し，その結果を返答するモデルを提案する．この一連の操作を 3 つのタスクに分割し，以下にその詳細を述べる．また，図 7 に本研究で提案するモデルのイメージを示す．

### 4.1 UI デザイン分類タスク

アプリケーションは様々な機能を提供し，ほとんどの場合は機能に合わせたいくつかの種類の UI デザインを含んでいる．このとき，UI デザインにおいて重要となる要素はその機能ごとに異なると考えられる．例えば，写真や動画を閲覧するひとつのアプリケーションの中でも，サムネイル表示を用いて項目

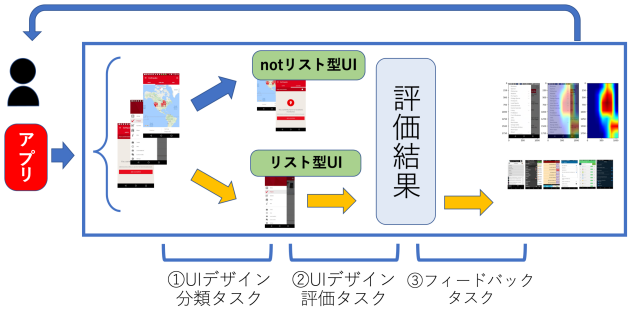


図 7　提案手法

を検索する場面と，実際に写真や動画を大きく表示して閲覧する場面とでは，求められるデザインは大きく異なる．このことから，UI デザインを評価するにあたっては，全く機能の異なる UI 同士の比較よりも，類似した機能を司る UI 同士を比較し，評価することが望ましいと考える．このため，UI デザインを類似したもの同士で分類するタスクが初めに存在すると考える．これを “UI デザイン分類タスク” と定義する．

本研究では，中でも，横長の選択項目が羅列される図 8 のような UI デザインを “リスト型 UI”，それ以外を “not リスト型 UI” と定義し，この分類・評価を研究対象として取り上げる．リスト型 UI は，多くのアプリにおいて項目選択や一覧の表示に広く用いられ，また多くの項目が並ぶことから，視認性やデザイン性が重要であり，アプリケーションのユーザビリティに大きな影響を与えると考える．このタスクの実現のため，データセット内の UI 画像に，人手でリスト型か not リスト型かのラベル付けを行い，そのラベルを用いて CNN を学習させ，分類モデルを生成する手法である．

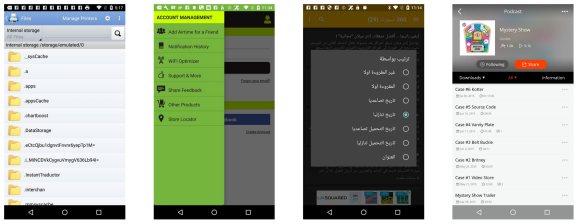

図 8　リスト型 UI

### 4.2 UI デザイン評価タスク

UI デザイン分類タスクを行った後，分類された UI 画面を，同一分類内の他の UI 画面と比較することでそれらを評価する．このタスクを “UI デザイン評価タスク” と定義する．本研究においては，リスト型に分類された UI のみを扱う．

このタスクの実現のため，アプリケーションのメタデータに含まれるユーザ評価を元に評価モデルを生成する手法を提案する．具体的には，初めにアプリケーションのメタデータとして提供される Play Store 内でのレーティングと，レーティング評価の投稿数をもとに，各 UI 画像を，評価の高い UI（本研究では “Good-UI” と定義），評価の低い UI（“Bad-UI”）に分類する．その後，この分類を教師ラベルとして各 UI 画像に付与し，UI

画像を説明変数，Good もしくは Bad のラベルを目的変数として CNN を学習させ，UI 画像を Good と Bad に分類する評価モデルを生成する．また，Grad-CAM を用いて，評価モデルの考察を行った．

### 4.3 フィードバックタスク

分類タスク，評価タスクによって得られた結果は，ユーザにフィードバック事となる．評価タスクでは Good もしくは Bad の予測を行ったが，ユーザがこのシステムを利用する場合，自身が入力した UI デザインの改善点の知りたいことが想定される．このとき，Good-UI であるか Bad-UI であるかの評価だけでなく，その根拠となる理由や，改善案を提示できることが望ましい．

本研究においては，この実現のため以下の 2 つを提案する．

第一は，UI レイアウトベクトルを用いた k 近傍探索をデータセット内で行い，入力 UI と類似すると思われる UI 画像のうち，良い手本となると考えられるものを提示する手法である．

第二に，Grad-CAM により，評価モデルがその評価を下すにあたっての根拠を可視化して提示する手法である．

## 5 実験・結果

前節で述べた提案手法により，3 つのタスクにおける実験を行った．以下に，実験の詳細と結果を述べる．

### 5.1 UI デザイン分類タスク

Rico から抽出した 6,000 件の UI 画像に対し，リスト型 UI，もしくは not リスト型 UI のラベル付けを行い，それらを教師ラベルとして UI 画像に付与し，CNN による学習を行った．ラベル付は 2 名により行い，それぞれが付与したラベルが一致したものを，最終的なラベルデータとした．リスト型に分類されたものが 1,672 件，not リスト型が 3,637 件，両者のラベルが一致しなかったものが 691 件であった．

リスト型 UI，not リスト型 UI とラベル付けされた合計 5,309 件の UI 画像について，70 %を訓練データ，30 %をテストデータとして CNN の ResNet18 [9] を用いて教師付き学習を行った．ResNet18 は，残差学習を用いた CNN であり，18 層で形成されている．ResNet18 のネットワーク構造のみを取得して利用したものと，ImageNet を利用して学習された pre-trained モデルをファインチューニングしたものとの 2 つのモデルで学習を行い，UI デザイン分類モデルを生成した．それぞれの学習の経過を図 9 に示す．（陽性：リスト型，陰性：not リスト型）．

また，ファインチューニングあり，なしのそれぞれのモデルの性能比較のため，ランダム化検定を行った．これはシステムが同一であると仮定した場合と比較して，実際の結果がどの程度珍しいかを示すものである．具体的には，比較するシステムにより出力された実際の評価値の平均の差と，システム間の同トピックをランダムにシャッフルして作成された B 個の結果の平均の差の分布とを比較する．p 値が 0.05 以下の場合は，有意水準 5 %においてシステム間の差が有意であることを示す．B=500 で行い，結果は $p < 0.01$ となり，2 つのモデルの差は統計的に有意であるといえた．

pre-trained モデルをファインチューニングし，10 エポックで学習を打ち切ったものを UI 画像分類モデルとし，本タスクで利用していない残りの 60,216 件の UI 画像について，リスト型 UI であるか not リスト型 UI であるかの分類を予測した．これによりリスト型と予測された UI 画像は，10,350 件であり，これらを抽出して次の UI デザイン評価タスクにおけるリスト型の UI 画像データセットとして用いる．

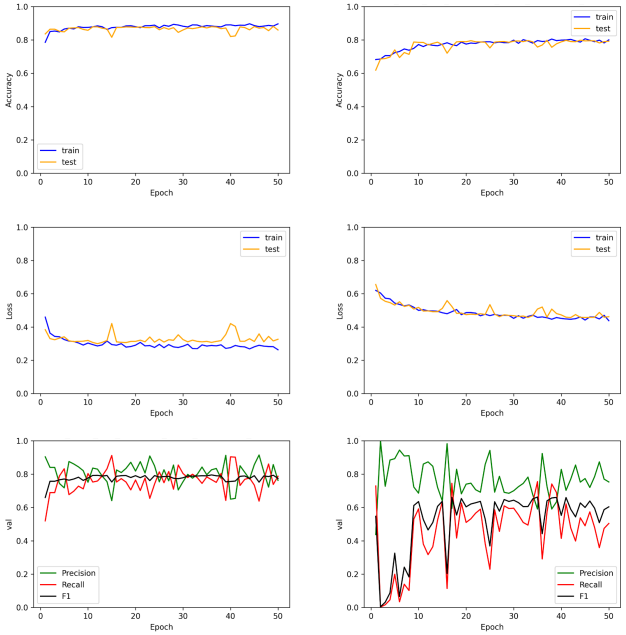


図 9　UI デザイン評価モデルの生成（左：ファインチューニングあり，右：なし）

### 5.2 UI デザイン評価タスク

前項で抽出したリスト型 UI 画像について，Rico に含まれるメタデータから，アプリケーションのレーティングと，レーティングの投稿数のデータを付与した．

このうち，レーティングが 4.5 以上かつ，レーティング投稿数が 10,000 件以上のものを評価の高い Good-UI，レーティングが 3.5 以下のものを評価の低い Bad-UI としてラベル付けした．リスト型 UI 画像に対するレーティングの分布を図 10 に示す．

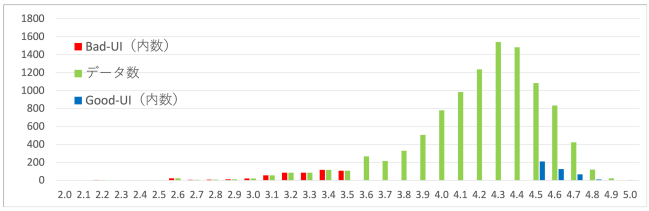


図 10　リスト型 UI におけるレーティングの分布

Good-UI に関して投稿数の条件を付加したのは，より多くのユーザに高い評価を受けている UI 画像を Good-UI として抽出するためである．

レーティングの分布を見ると，レーティング＝ 3.6 の件数が前後より高くなっていることがわかる．この事から，レーティング 3.6 を超えられるか否かが 1 つの基準になると考えた．また，ユーザの少ないアプリケーションは，ユーザビリティの低さがその要因の一つと考えられるため，Bad-UI に関して投稿数による条件を付加していない．Good-UI とラベル付けされたデータは 411 件，Bad-UI は 524 件であった．

Good-UI と Bad-UI 計 935 件のうち, 70 %を訓練データ, 30 %をテストデータとして学習を行い，UI デザイン評価モデルを生成した．学習においては，ImageNet を利用して学習された pre-trained モデルをファインチューニングした．結果を図 11 に示す．

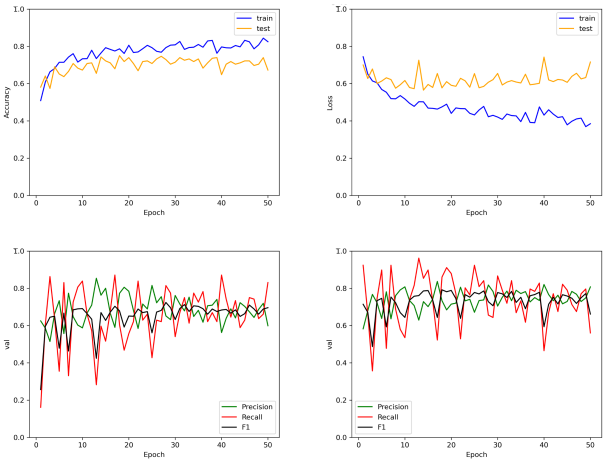


図 11　UI デザイン評価モデルの生成（左下：Good-UI が陽性，右下：Bad-UI が陽性）

## 5.3　フィードバックタスク

### 5.3.1　レイアウトベクトルによる k 近傍探索

Bad-UI 改善例の提示のため，Rico 内のデータでの k 近傍探索を行い，その結果を確認した．この手法では，Bad-UI のレイアウトベクトルを用いて， Bad-UI 自身と同じアプリケーションの画像を除いたものの中で近傍上位から順に，以下の各条件により UI 画像を抽出する．一例を以下に提示する．左端が入力された UI 画像であり，右に行くほどレイアウトベクトル空間内で遠いものである. 各項目が近傍探索において上位何件目だったかと，そのアプリケーションのレーティングを [近傍探索順位：レーティング] という形で示す．つまり，例えば図 13 での “258 位:4.6” とは，Good-UI に含まれるものという条件下で近傍探索を行った場合に，最近傍であった画像が，条件を設けない全画像での探索では 248 番目に近傍であり，そのアプリのレーティングはメタデータを参照することで 4.6 であったことを示している．

条件 1：最近傍であるもの

条件 2：Good-UI に含まれるもの

条件 3：リスト型であり，かつアプリケーションのレーティングが 4.6 以上のもの

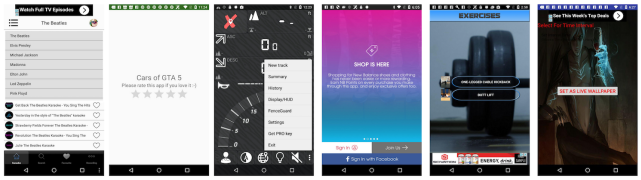

図 12　最近傍 [近傍探索順位：レーティング][1 位：4.1，2 位：4.3，3 位：4.0，4 位：4.2，5 位：4.2]

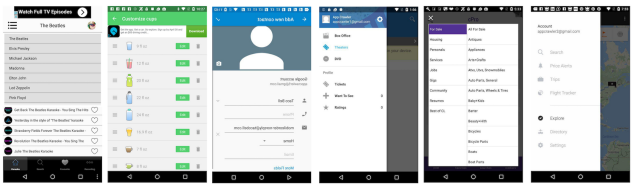

図 13　Good-UI 条件 [248 位：4.6，686 位：4.6，687 位：4.6，702 位：4.5，741 位：4.5]

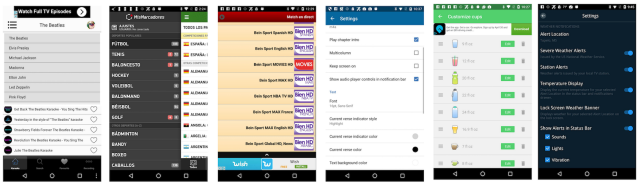

図 14　リスト型かつレーティング 4.6 以上 [51 位：4.6，104 位：4.6，187 位：4.7，248 位：4.6，306 位：4.6]

### 5.3.2　Grad-CAM

システムからユーザに対して行うフィードバックを想定し，Grad-CAM を用いて UI デザイン評価モデルの，予測根拠の可視化を行った．Good-UI を陽性として，図 15，16，17，18 にその例を示す．ヒートマップが赤くなっているエリアほど，予測に強く影響したことを示している．なお，ヒートマップの状況を確認しやすいように，左部に元々の UI 画像，右にヒートマップ要素だけを取り出したもの，中央にそれらを重ねたものを示す．

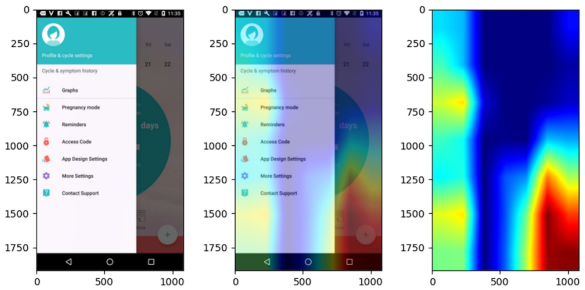

図 15　UI デザイン評価タスクにおいての TP 例

# 6　考　　察

## 6.1　UI デザイン分類タスク

はじめに，リスト型 UI の抽出を目的とした UI デザイン分類タスクの各手法について考察する．

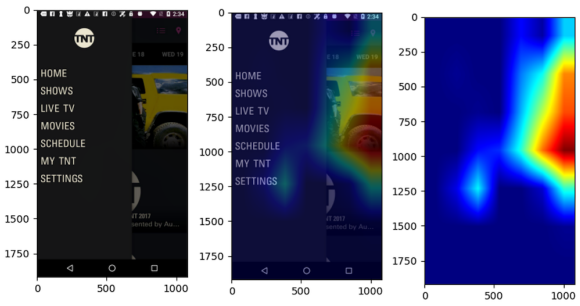


図 16　UI デザイン評価タスクにおいての FP 例

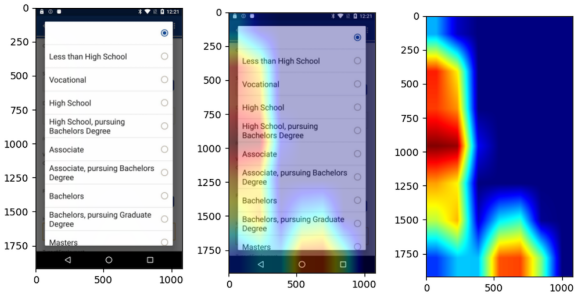


図 17　UI デザイン評価タスクにおいての FN 例

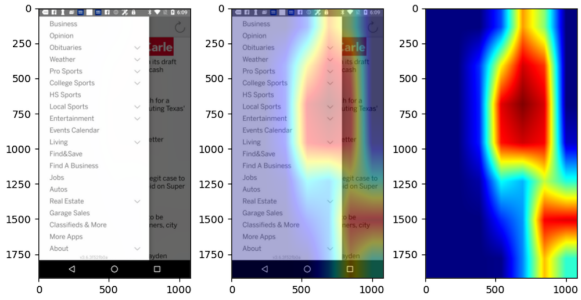

図 18　UI デザイン評価タスクにおいての TN 例

### 6.1.1　ファインチューニング

本実験では，ResNet18 において，モデルの形状のみを取得して利用したものと，ImageNet を利用して学習させた pre-trained モデルをファインチューニングしたものとの両方を利用した．

ファインチューニングを行ったモデルでは学習初期の段階で各評価指標が高い状態で安定する結果となった．

ImageNet は自然画像を収集したデータセットであり，本研究で分類対象とした UI 画像とは大きく異なるものである．しかし今回の結果から，UI 画像の分類タスクにおいても，効果的な学習を行う上でファインチューニングが有効な手段であると考えられる．

### 6.1.2　評 価 指 標

以下では，リスト型 UI を陽性，not リスト型 UI を陰性として議論する．

本項目における分類タスクは，UI 画像をリスト型（陽性）と，not リスト型（陰性）とに分けるものであり，用いた評価指標は Accuracy，Precision，Recall，F-1 である．分類タスクにおける教師ラベルは，リスト型が 31 %，not リスト型が 69 %であった．この事から，多数である not リスト型 UI が正しく分類できてさえいれば（真陰性が多ければ），陽性のデータの分類が不十分でも Accuracy が高いスコアとなる．本来であれば not リスト型は更に細分化して分類されるべきであり，見かけ上 Accuracy が高いモデルであっても，UI 画像を多クラスに分類するタスクにおいて精度を保てるか否かについては再度検証を行う必要があると考えられる．

Precision はリスト型と予測した分類がどれだけ正しいか，Recall はどれだけ多くのリスト型を網羅して分類できているか，F-1 はその 2 つを総合的に評価できる指標といえる．仮に Recall が高く Precision が低いモデルがあった場合，これはリスト型 UI を多く抽出できるが，偽陽性であるデータを多数陽性と分類してしまうものである．対して Recall が低く Precision が高いモデルは，陽性の抽出率は低くとも，リスト型 UI をより正確に抽出できるモデルと考えられる．

実用上は，ユーザがアプリケーションを入力すると，そこに含まれる UI 画像を抽出・分類した後，それらを評価しフィードバックを行うシステムが考えられる．この場合，システムが分類した UI 画像を一旦ユーザに提示し，正しく分類できていない画像についてはユーザが修正して評価するという方法が考えられる．この方法を取る場合は，システムの分類タスクは，ユーザの作業を短縮する目的で行うと考えられるため，Accuracy が高いモデル，もしくは大きな割合を占める分類の UI 画像に対する Recall が高いモデルが望ましいと考えられる．

### 6.1.3　Grad-CAM を用いた分析

分類タスクを行った後，Grad-CAM を用いて，分類モデルが予測を行う上で重視する画像のエリアの可視化を行った．

図 19 では，リスト型とラベル付けされた画像を正しく判別できている例を示す．画面半分の，項目が区切られている部分を認識することで，リスト型 UI と予測できていると言える．

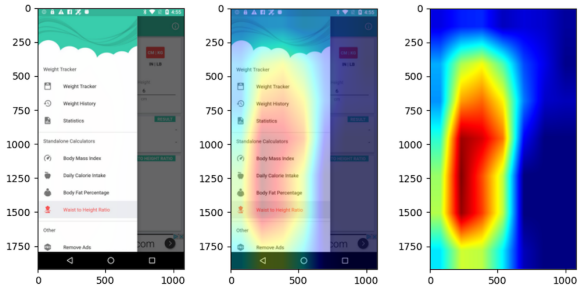

図 19　UI デザイン分類タスクにおいての TP 例

図 20 では，リスト型の教師ラベルが付与されているが，モデルが正しく予測できなかった例を示す．これらの画像を正しく分類できなかった理由として考えられる要因を 3 点挙げる．第一に，分類モデルが注目している範囲が，リスト型の根拠となる部分から外れてしまっていることである．画面上部に注目し，リストが表示されている画面下部への注目度は低いことがわかる．第二に，学習のための処理として画像を圧縮したことで，必要な情報が失われていることである．UI 画像では，画面下部は項目ごとの区切りとして，細い横線が引かれている．しかし，画像を圧縮処理したことで，分類器が識別できない要素となってしまった可能性がある．第三に，注目した要素がリス

ト型の特徴と大きく異なることである．正しく分類できた例に示したとおり，リスト型 UI の分類において重要な要素は，画面内で多くの横線で区切られた範囲を認識できることであると考えられる．図では，画面上の縦線に注目してしまっており，その特徴はリスト型と大きく異なっている．

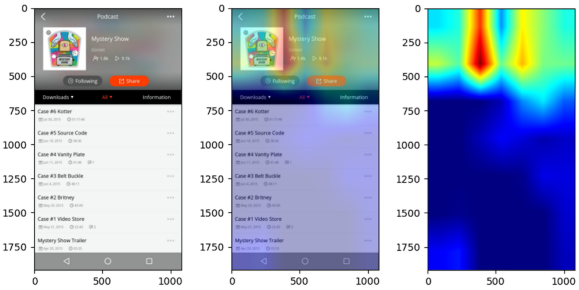

図 20　UI デザイン分類タスクにおいての FN 例

図 21 では，not リスト型の教師ラベルが付けられているが，リスト型と予測した例を示す．今度は対象的に，画面右端の，横線が数本存在するエリアに注目している事がわかる．学習モデルは画面内の要素を平面的に捉えており，人間が知覚する，画面上の立体的構造，奥行きを捉えられていないと考えられる．

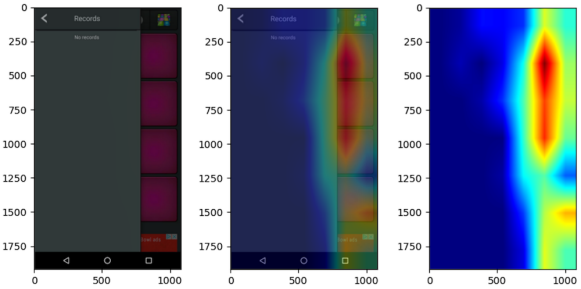

図 21　UI デザイン分類タスクにおいての FP 例

### 6.2　UI デザイン評価タスク

本実験では，UI 画像のみを説明変数として用い，CNN に学習させることで Good-UI と Bad-UI とを評価するモデルを提案した．Accuracy が 0.5 を上回っているものの，十分高い性能であるとは言い難い．この要因について考察する．

第一に，十分な分類精度を出すに必要なデータ量でなかったことが考えられる．本実験においてはリスト型 UI に分類された UI 画像の内，ユーザによる評価が高いと考えられるものを抽出し，それらを用いてファインチューニングを行った．このときの学習におけるデータ量が十分ではなかった，もしくは，何らかの要素について偏りのあるものであったことが考えられる．Rico は，Play Store の各カテゴリ内ランキングにおいて上位 200 件に入っているアプリケーションからデータを収集しており，またデータセット内の平均レーティングが 4.1 である．このことから，比較的ユーザビリティの高い UI を多く含むデータセットであることが考えられる．この点の検証には，更に大きなデータ量を用いた学習を行った場合との比較が必要であると考える．

第二に，UI 評価タスクを行う以前での．リスト型 UI 分類タスクの精度が不十分だったことが考えられる．本研究ではリスト型 UI 分類モデルにより分類されたものを全てリスト型 UI である前提で評価タスクを行っている．分類されたものの中に一定数の偽陽性データが含まれていると考えられる．このデータを用いたことで，効果的な学習が阻害されてしまった可能性が考えられる．

#### 6.2.1　Good-UI・Bad-UI のラベル付け根拠

本研究では，Play Store でのユーザによるレーティングとその投稿数を，ユーザビリティ評価の指標として用いた．しかし，これらの要素はアプリで提供されているサービスの内容や，動作の安定性，流行によるユーザ数の増減，投稿を行うユーザ層か否かといた，UI デザインとは異なる部分に影響を受けるものと考えられる．UI デザインの自動評価においては，定量的にユーザビリティを評価できる要素について，さらなる検討が必要と考える．

#### 6.2.2　低評価抽出と高評価抽出

本実験においては，低評価 UI を陽性とした場合と，高評価 UI を陽性とした場合との両方の精度を算出した．これは，評価モデルを利用する目的によって重視すべき点が異なると考えられるからである．

例えば，以下のように異なる目的で UI 評価タスクを行うことが考えられる．

まず，アプリケーション作成者に，そのアプリケーションの中で改善すべき UI を提示するシステムを構築する場合である．この事例においては，Bad-UI の認識精度が重要となる．元々ある程度優れた UI を判断できることよりも，改善の必要がある UI を指摘できる機能が重要と考えられる．

反対に Good-UI の認識精度が重要となるケースとして，優れた UI のデータベースを生成し，それを利用する場合など，Bad-UI の利用を必要としない場合である．

### 6.3　フィードバックタスク

本研究では，モデルからユーザへのフィードバックとして，k 近傍探索を行い，入力された UI 画像と類似したものを抽出し，提示する手法と，Gard-CAM によるヒートマップでの出力を検討した．これらについて考察を行う．

#### 6.3.1　k 近傍探索

最近傍の 5 件を抽出した条件においては，入力された例と類似しているとは言い難い UI 画像が抽出された．レイアウトベクトルが提供するベクトル間での差異の大小は，人間が知覚して判断する類似度とは乖離する場合があるといえる．

Good-UI に限定した条件での抽出は，本研究で生成した UI 評価モデルを用いたフィードバックであると言える．結果は前節に示した通りである．最近傍を探索した例よりも直感的に類似していると感じやすい UI 画像が提示された．また，全データ内に対する近傍探索で何番目に近かったかという順位については，200 位を超えた順位となっている．生成した Good-UI 群に限定した抽出したためであるが，この間にも多数の有用な UI デザインが存在すると仮定すると，必要以上に厳しい抽出条件

であった可能性がある．また，多数の例を確認したい場合でも，Good-UI に分類された UI 画像数が上限である．

このため，UI 分類タスクにおいて生成したモデルにおいてリスト型と予測され，レーティングが 4.6 以上であるという条件で抽出を行った．この手法は，UI 評価タスクをレーティングに委ね，本研究で生成した UI 分類モデルを用いた手法であるといえる．Good-UI のみに絞った手法よりも近傍順位が上位である，つまりレイアウトベクトル空間上では近い位置に存在する画像を抽出できている．

但し，前述の通りこの結果は本研究で生成したリスト型 UI 分類モデルを用いているため，精度もモデルに依存したものとなる．また，リスト型 UI 以外の探索は行うことは出来ない．

更に，今回の実験では，ではどのような抽出がなされるかを確認したのみであり，有用性や実際に利用した場合の評価は行っていない．抽出した UI が入力と類似しているかの判断も，ユーザにより大きく異なると考えられる．

**6.3.2** Gard-CAM による提示

UI 分類タスクの考察でも用いた Grad-CAM により，UI 評価モデルが画面上のどの部分を重視して予測を行ったかを可視化した．モデルが重要視したエリアの，中心地点の位置や，その分布範囲，分布の形状や，画面上の構成要素や色等を観察したが，根拠とする要素の解釈を行うことは出来なかった．このことから，本タスクにおいて Grad-CAM をフィードバックタスクの有効な手法として利用することは出来なかった．

## 7 結　　論

本研究は，システムに入力された UI 画像の評価について，UI デザイン分類タスク，UI デザイン評価タスク，フィードバクタスクの 3 つに分割し，それぞれのタスクにおける手法を提案した．UI デザイン分類タスクにおいては，ファインチューニングが有効な手段であることを示し，さらに Grad-CAM を用いることで，モデルがどのように UI 画像を分類しているかにつての可視化を行った．また，UI デザイン評価タスクにおいては，アプリケーションのメタデータを用いることでラベル付けする手法を用いて，評価モデルを生成した．フィードバックタスクにおいては，K 近傍探索を用いて構造的に類似した UI 画像を抽出する中で，分類モデルと評価モデルを併用する手法を検討した．

## 8 今後の課題と展望

UI 分類タスクにおいては，本研究で扱うことの出来なかった，リスト型以外の分類項目の設定と，その実施が課題である．評価タスクにおいては，ラベル付けに用いる指標の再検討と，精度の向上が課題である．

また，本研究における UI 画像の分類は，画面上の要素をもとにした，レイアウトベースの方法であったと考えられる．他の分類方法として，各 UI 画像が目的とするタスクベースでの分類方法が考えられる．例えば，必要事項を入力して次の画面に進むという機能が同じ場合でも，日常的に使うサービスのログイン画面と，重要な契約の同意画面とでは，設計意図に違いが生じる．これらの画面のレイアウトは類似したものであることが予想されるが，ユーザが内容に同意するボタンが，押しやすい場所にわかりやすい形で置かれているものと，操作性は劣るものの，押し間違いが少ないように設計されているもののと評価は，必要とされる場面に応じて分けて考えられるべきである．「目的の項目をタップする速度」や，「タップ操作の正確性」といった，定量的に評価可能である指標を用い，レイアウトごとに（もしくはレイアウトを横断して），タスクを考慮してさらなる分類・学習を行うことで，より現実に即した分類・評価を行うことが考えられる．

フィードバックタスクにおいては，提示方法の再検討が課題であり，よりユーザに対する具体的なデザイン改善の手立てとなる内容が望ましい．例えば，ユーザの入力が，UI デザインのアンチパターンに類似した場合，その改善例を提示する手法が考えられる．

## 文　　献


[1] https://material.io/design

[2] Z. Jiang, R. Kuang, J. Gong, H. Yin, Y. Lyu and X. Zhang, “What Makes a Great Mobile App? A Quantitative Study Using a New Mobile Crawler,” 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE), Bamberg, 2018, pp. 222-227, doi: 10.1109/SOSE.2018.00037.

[3] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A Mobile App Dataset for Building Data-Driven Design Applications. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST ’17). Association for Computing Machinery, New York, NY, USA, 845 – 854. DOI:https://doi.org/10.1145/3126594.3126651

[4] Biplab Deka, Zifeng Huang, and Ranjitha Kumar. 2016. ERICA: Interaction Mining Mobile Apps. In Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST ’16). Association for Computing Machinery, New York, NY, USA, 767 – 776. DOI:https://doi.org/10.1145/2984511.2984581

[5] Forrest Huang, John F. Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based User Interface Retrieval. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI ’19). Association for Computing Machinery, New York, NY, USA, Paper 104, 1 – 10. DOI:https://doi.org/10.1145/3290605.3300334

[6] Amanda Swearngin and Yang Li, “Modeling Mobile Interface Tappability Using Crowdsourcing and Deep Learning”, CHI ’19: Proceedings of the 2019 CHI Conference on Human Factors in Computing SystemsMay 2019 Paper No.: 75 Pages 1 – 11https://doi.org/10.1145/3290605.3300305

[7] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, D, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization”, International Journal of Computer Vision, 2, Oct pp. 336 – 359, 2019.

[8] Thomas F. Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomir Mech, and Ranjitha Kumar. 2018. Learning Design Semantics for Mobile Apps. In Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST ’18). Association for Computing Machinery, New York, NY, USA, 569 – 579. DOI:https://doi.org/10.1145/3242587.3242650

[9] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.