

深層学習を用いたタイトルロゴの自動生成システム 「LogoLabo」の実装

松永 康太¹ 杉浦 優聡¹ 小瀬 将史¹ 渡邊 一樹² 濱川 礼¹

概要: 本論文では、タイトルの文字列からフォントの生成と着色を行い、タイトルロゴの自動生成するシステム「LogoLabo」について述べる。世の中には様々なロゴが存在するが注目を引く効果的なロゴの作成には時間が必要である。「LogoLabo」では深層学習を用いることでタイトルロゴの自動生成を行い、時間短縮を図る。タイトルロゴ生成は文字のフォント生成と画像変換によって行う。我々は文字の形をした果物を含むタイトルロゴを魅力的と考え、今回は果物での実装を行なった。

1. はじめに

論文や資料等にはほとんどの場合タイトルが定められており、読み手が初めに目を通す部分である。故に、タイトルには筆者が伝えたいことを読み手にいち早く印象付ける効果があり、「その資料を読みたいか否か」という部分に対して大きな影響を与える [1]。また、コンセプトやイメージといった情報を伝え、印象に残りやすい効果をもつものとしてロゴは研究が進められている [2][3]。ロゴにはいくつか分類方法があり、例えば文字列のみをロゴタイプ、図柄をシンボルマーク、ロゴ全体をロゴマークとしたものがあげられる [4]。タイトルを考える際には文字列を扱うためフォントについて考えなければならない。フォントには印象によって商品選択等に影響を与えることがありと先行研究にて示されている [5][6]。また、フォントが影響を与えることから、よりユーザの思うようなフォントを生成しようと研究が進められている [7]。

本論文では、印象付けるのに重要な役割を持つタイトルの文字をロゴマークとして扱うタイトルロゴに注目した。前述の分類に従うと、タイトルロゴは文字列を表すフォントとデザイン性の強いシンボルマークからなる。工夫されたタイトルロゴを作成するにはデザインセンスが問われる。しかしそれは、先天的な感覚や経験に左右されるため、簡単に身に付ける事が出来ない。なにかを伝えたいと考えた時に、いくつかのフォントの中から適切なものを選ぶという行為にも同じ事が言える。

従って、工夫されたタイトルロゴを作成しようと考えた場合、デザインセンスを身につけていない人が自分で作成

するのは困難だと考えられる。自分で作成出来ないならばデザイナーに依頼するのが一般的である。それには依頼料はもちろん納期の決定、必要な情報の受け渡し、完成イメージを伝達、そして完成したデザインに対して修正があればまた時間がかかる。このように金銭的・時間的な理由から、気軽にデザイナーに依頼することもまた困難である。この問題を解決するため、我々はタイトルロゴ自動生成システム「LogoLabo」を実装した。事前にユーザに対してイラスト化したい物体とフォントについての印象評価を実施し、その結果を基に使用するフォントを決定する。その後、イラストに変更したい文字を選択してもらい最終的な出力を行う。

2. 関連研究

個人の印象をフォントに適応させたり文字の形を変える、また着色を行う研究は盛んに行われている。本研究に類似した研究例について述べる。

2.1 画像の感性を反映させたフォントの自動生成手法

中村らは、人の感情を画像から抽出しフォントに適応させる研究を行っている [8]。感情の抽出には Microsoft 社が提供している感情検出 API である Emotion API を用いており、写真から感情を認識させてラベル付けを行っている。検出する感情は「喜び」「悲しみ」「怒り」「嫌悪」「恐怖」「驚き」「軽蔑」「無表情」の合計 8 種で、それぞれの感情を表す確率を算出している。また、上記 8 種の感情を順に示し、各感情に最もふさわしいと感じるフォントを 9 つから 1 つずつ選択する形で、感情とフォントに対する印象調査を行いフォントとの対応付けを行っている。フォントの生

¹ 中京大学 工学部 情報工学科

² 中京大学 工学研究科 情報工学専攻

成については、pix2pix を拡張させたモデルである zi2zi[9] が用いられている。しかし、感情のない物体をフォントと対応付けることは出来ないため「LogoLabo」では形容詞対の質問を行い、フォントに対応付けている。図1でこのシステムの出力例を示す。



図1 感情を推定してフォントを生成した例

2.2 ユーザの持つ印象を反映した対話型ロゴデザインシステム

鬼沢らは、画像挿入によってアルファベットの一部分と画像との入れ替えを行ない、画像のような文字を実現している [10]。画像挿入を行う方法としてアルファベットの辺ごとに画像に置き換えた組み合わせをユーザに提示し、ユーザが満足するロゴデザインを選択後、挿入画像に対して微調整を加えることができるようになっている。例えば、コウモリのシルエット画像をアルファベットの A の一部分に対して置き換える。ユーザは専用のインターフェースを使って、どの辺に置き換えた画像がいいかを選ぶことでコウモリと A が組み合わさった画像が完成する。実際の出力例を図2に示す。しかし、変換できるのは画像の一部分であり、文字全体を画像に対応した変換は行われていない。さらに、アルファベットと文字の組み合わせの選択や用意した画像の変換はユーザが選択する手間がある。そこで我々の「LogoLabo」では、文字の輪郭画像から深層学習によって果物のように自動着色を行うことで、文字全体の変換と自動生成によりユーザの手間を軽減している。



図2 コウモリと A が組み合わさった画像例

2.3 GANime

Tai らは、線画からフルカラーへの着色に対し定量的な評価から複数のモデルの検討を行なっている [11]。評価の対象となるモデルには Neural Style Transfer, pix2pix, CycleGAN の三つとなっており、定量的な評価方法として Structural Similarity(SSIM) と Frechet Inception Distance(FID) を用いている。評価結果を示す表1から pix2pix は FID と SSIM の標準偏差が最も低く、平均 SSIM が最も高いことがわかっている。つまり、SSIM の標準偏差から入力画像の複雑さに対するロバスト性が優れており、また、FID と平均 SSIM から全体的な画質が最も高いことが示されている。このことから我々の「LogoLabo」では pix2pix を用いて自動着色を行う。

3. 提案手法

事前にフォントやイラスト化したい物体に関する印象評価を行う。これに対するユーザの回答を基にフォントの生成を行い (3.1), フォントのエッジ画像からイラストを生成する手法 (3.2) を提案する。

3.1 フォントの生成

5種のフォントとイラスト化したい物体に対して、表2に示す25個の形容詞対を用いた7段階のリッカート尺度で印象評価を行う。フォントには、図3に示す既存フリーフォントの「851 テガキカクット」[12]「font1」[13]「源暎ゴシック P ExtraLight」[14]「Nishiki-teki」[15]「装甲明朝」[16]を用いた。後の印象調査でコサイン類似度に違いが出るように特徴の異なるフォントを、フォントをまとめているサイトから引用した [17][18][19]。

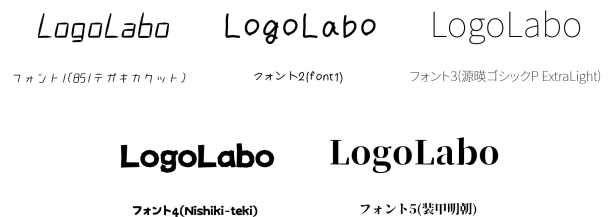


図3 評価対象の5種のフォント

3.1.1 フォントの選定

ユーザの回答を1次元ベクトルとし、 \vec{q} をフォント、 \vec{d}

表 1 三つのモデルの検証結果

Model	FID	SSIM(mean)	SSIM(standard deviation)
Neural Style Transfar	345.506	0.6547214	0.09885219
Pix2Pix	227.948	0.7468922	0.07413621
CycleGAN	272.619	0.7238495	0.08240251

表 2 25 の形容詞対

やわらかい - かたい	明るい - 暗い
好きな - 嫌いな	美しい - 醜い
澄んだ - 濁った	暖かい - 冷たい
湿った - 乾いた	はっきりした - ぼんやりした
丸い - 四角い	派手な - 地味な
洋風 - 和風	贅沢な - 質素な
複雑な - 単純な	子供っぽい - 大人っぽい
男性的な - 女性的な	カジュアル - フォーマル
陽気な - 陰気な	迫力ある - 物足りない
軽い - 重い	さっぱりした - 濃厚な
渋い - 華やかな	繊細 - 豪快
甘い - 甘くない	安定した - 不安定な
大きい - 小さい	

を物体への回答とした際のコサイン類似度を以下の式で求める。

$$\cos(\vec{q}, \vec{d}) = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \cdot \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

5種のフォントそれぞれに対して物体とのコサイン類似度を算出する。既存のフォントには限りがあり、コサイン類似度が1位のフォントだけでは十分に対象となる物体を表現出来ないと考えた。コサイン類似度2位のフォントにも物体を表現するために有効な要素が多く含まれており、コサイン類似度が1・2位のフォントをブレンドさせることで、より良く物体を表現したフォントが生成出来るのでは無いかと考え zi2zi を使用した。

zi2zi は Yuchen らが提案したモデルで、深層学習の一種である Generative Adversarial Network(GAN) を用いた pix2pix モデルを拡張し、漢字に適用させたものである。GAN とはランダムなノイズベクトル z から出力画像 y への関係を学習する生成モデルである。Conditional GAN(CGAN) では条件画像 x とランダムノイズベクトル z から画像 $G(x, z)$ を生成する。生成器 G は敵対的に訓練された識別器 D によって本物と区別できない出力を生成するように訓練され、識別器 D は生成器の偽物を可能な限り検出されるように訓練される。この手順を図4に示す。pix2pix では CGAN をベースとしている。

zi2zi ではある文字を指定した別のフォントの文字へ変換させたり、2つのフォント間で補間が行える。本研究では zi2zi の補間を用いて2つのフォントをブレンドさせる。

3.1.2 ブレンド割合の決定

求めたコサイン類似度を基に、コサイン類似度2位のフォントのコサイン類似度の比率が足して1になるように

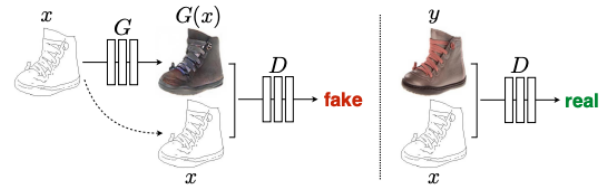


図 4 GAN の構造 [22]

調節し、ブレンドの割合を決定した。

3.2 着色手法

zi2zi によって生成された文字の着色を pix2pix で行う。pix2pix とはペア画像間の関係を学習し、画像着色や画像補完が可能な手法である。本研究では入力側を学習対象である果物の輪郭画像、出力を果物の着色済み画像でペアを作り学習を行なった。この入出力を図5に示す。入力に対応する輪郭画像は opencv で実装されている cv2.Canny によって作成できるエッジ画像を使用する。出力に対応する果物の着色済み画像にはエッジを取りやすくするために、k-means 法を用いて減色処理を行いイラスト化する。一枚の画像からエッジ画像と減色処理を行なった画像を作成しペア画像とした。

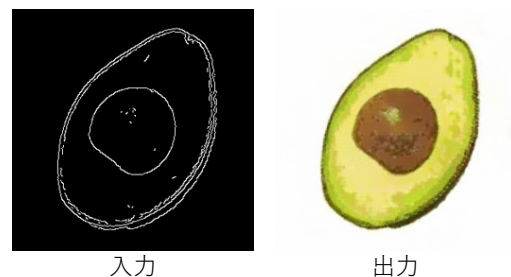


図 5 エッジ画像と着色画像のペア

4. システム実装

「LogoLabo」はユーザーの入力した文字をもとにタイトルロゴ生成を行う、タイトルロゴ自動生成システムである。システムの流れは図6に示す。ユーザーがフォントと果物に対する印象を各25項目7段階で入力し、印象のデータをもとにフォントの生成を行う。我々は図7に示すような、ユニリーバ社の所有するリプトンブランドロゴ [20][21] に入っている文字の形をした果物画像を魅力的だと考え、今回は文字の形をした果物画像を含むタイトルロゴ生成に焦

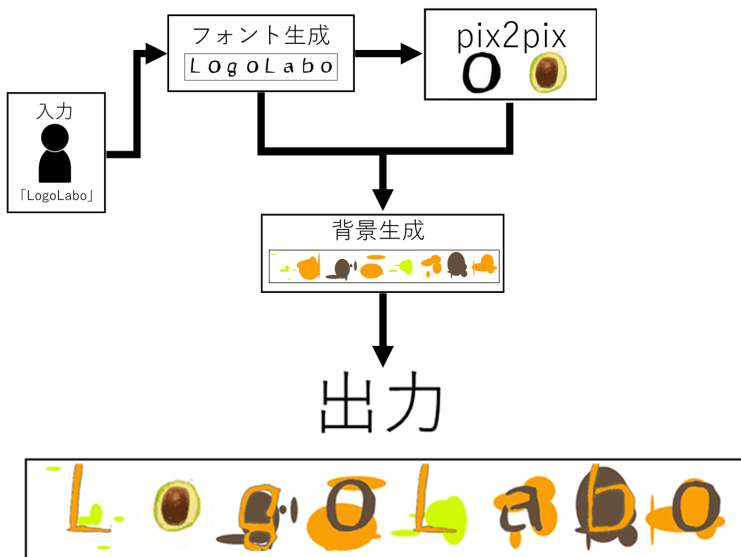


図 6 システムの流れ



図 7 ユニリーバ社が保有するリプトンブランドのロゴマーク

点を当てた。生成したフォントの一字を選び深層学習を用いて果物に近い着色を行う。生成されたフォントと着色されたフォントを結合させ、さらにランダムに生成した背景の画像を合成し表示する。以上の機能を持った GUI アプリケーションの開発を行った。

4.1 フォント生成

4.1.1 学習

学習にはフォント毎に属性ラベルを付与した画像データセットが必要となる。データセットには TTF や TTC 形式のソースフォントとターゲットフォントを用いて、JPG 形式で画像データ化したものを扱う。フォントはペアで扱いそれぞれに属性ラベルを付与させる。Windows に標準搭載されている「BIZ UD 明朝 中」をソースフォントとし、前述の 5 種のフォントをターゲットフォントとした。フォントによって扱う漢字に大きく差があるため、ひらがな 80 文字、カタカナ 80 文字、アルファベット 52 文字、数字 10 文字とした。画像サイズは 256 × 256 pixel で計 1110 枚、batchsize を 8、epoc 数を 200 で学習を行った。訓練用とテスト用のデータ比率は 9:1 である。

4.1.2 推論

学習したモデルを使用し、ランダムな文字列を推論させた結果を図 8 に示す。また、2 つのラベル間の比を徐々に

く	け	ふ	う	そ	B	U	t	n	A
ヶ	ぜ	る	ど	ア	R	o	e	J	∩
ダ	ズ	ョ	ガ	わ	w	F	Z	J	b
パ	い	た	エ	プ	l	c	v	Y	N
ゃ	ソ	リ	ハ	ア	r	l	x	X	E
シ	イ	フ	め	d	L	h	W	p	s
へ	オ	ヲ	ひ	H	f	M	u	y	C
ル	に	な	ト	T	K	g	a	P	m

図 8 ランダムな文字列の推論例

変化させ補間させた例を図 9 に示す。zi2zi は元々漢字用に提案されたモデルであるものの、結果を見ると仮名やアルファベットもおおよそ学習出来ていることが確認出来る。

4.2 pix2pix 実装

Google の画像検索を用いてアボカドの種が写っているカット済み画像を 125 枚収集した。収集したデータを 1 枚につきランダムに 5 種類の角度変換を行うことで、630 枚に水増ししたものを学習データとして使用した。入力収集したデータから生成したエッジ画像とし、出力はエッジ画像に対応した減色処理済みの着色画像とする。訓練用データとテスト用データに 9 対 1 の割合で分割し学習を行なった。学習方法は batchsize を 1、epoc 数を 400 とした。epoc 数を 1000 にしての出力も行なったが、アボカドの種の色が濃く出てしまったため epoc 数は 400 での実装とした。出力結果を図 10 に示す。

4.3 背景生成

背景生成ではランダムに生成される、三角形・四角形・円を最大 9 色まで色を変化させることができるが、初期状態は果物画像に含まれる上位 3 色を抽出している。上位 3 色の抽出は画像を RGB から HSV 変換させ、色相を 9 段階に分け画像に対する色面積の多いものを選んでいる。白や黒などは明度や彩度によって面積には合計されないようにした。生成される図形は python の pillow を使い、画像の縦・横幅から 50 分の 1 以下で生成されないようになっている。

4.4 GUI アプリケーション

「LogoLabo」の GUI アプリケーションは python の標準

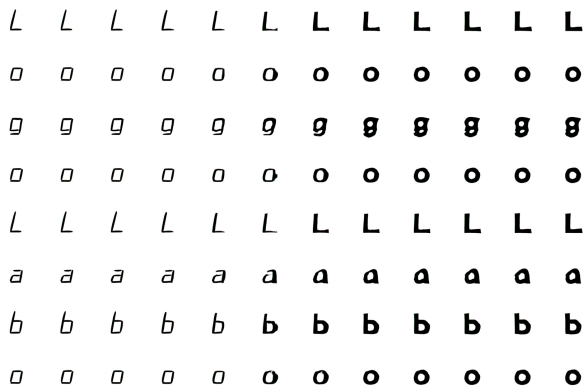


図9 フォント間を12ステップで補間させた例

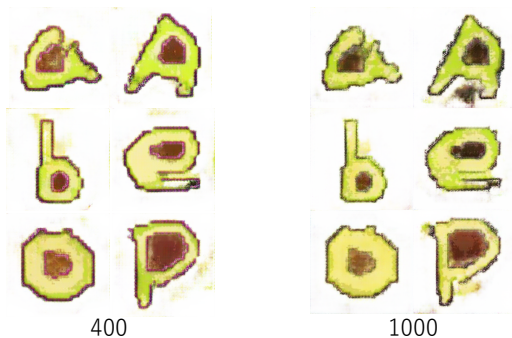


図10 出力結果

インターフェースである tkinter を用いて製作した。「LogoLabo」にはアンケート・文字入力・変換文字選択・背景生成の4つの画面がある。アンケート画面ではフォント生成時に使用するテキストデータの保存を行っている。文字入力画面で文字のデータを取得した後、フォント画像を生成する。変換文字選択画面でアボカドに変換する文字を選択し、フォント画像に着色を行い各画像を結合する。最後に現在のロゴ画像を表示し、ボタンを押すことでランダムに背景が生成される。

5. 評価実験

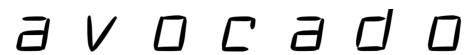
評価者には「LogoLabo」を使ったロゴ生成と評価者自身でロゴを生成する2通りを行って頂いた。評価者自身のロゴ生成例が図11である。また「LogoLabo」を使用した後にアンケートをとった。アンケートは「文字がアボカドに変化しているのか」、「アボカドに沿ったフォントか」、「『LogoLabo』は使いやすいのか」、「『LogoLabo』の自動生成と手動での時間の差」、「図12に示す既存のロゴ[23]と比べて生成したロゴを使いたいと思ったか」、「『LogoLabo』を再度使用したいか」の6項目5段階に加え、文字が一部アボカド化された画像、コサイン類似度より選出されたフォントのみの画像、「LogoLabo」で生成した背景付き画像の3種類に順位付けをする1項目を足した、計7項目を検証した。なお、3種類の画像は図13の(a)(b)(c)に一例を示した。また、「LogoLabo」でロゴを生成する場合とほ



図11 評価者の生成例



図12 Wholly Guacamole社のロゴマーク



(a) フォントのみの画像



(b) 背景なし生成画像



(c) 背景付き生成画像

図13 出力画像例

かの手法でロゴを生成する場合の時間計測を行った。「アボカドに沿ったフォントか」の項目に関しては、5つのフォントとアボカドのアンケート結果からコサイン類似度を求めた。

5.1 評価収集方法

「LogoLabo」の評価方法は、ユーザーがアプリケーションを使用した際の印象をアンケートで収集する方法より評価を行った。データは大学生13人より収集した。ロゴ生成に関しての時間の計測は13人のうちの5人を対象に行った。5つのフォントとアボカドに対してのコサイン類似度は時間計測で対象にした5人のデータをもとに算出した。

アンケート結果は表3-8で「そう思う」を5として1-5で平均を取ったものが表9である。表3・表7・表8より「文字はアボカドに変化しているか」、「再度『LogoLabo』を使いたい」、「自分で生成する方法と比べ早く生成できたか」の項目は評価が高いことがわかる。しかし、表4、表6より「『LogoLabo』は使いやすいか」、「既存のロゴと比べてこのロゴを使いたいか」の項目では、平均が3を下回る結果となった。

5.2 結果

図14はフォントのみのロゴ(例:図13a)と「LogoLabo」

表 3 文字はアボカドに変化しているか

	人数
そう思う	2
少しそう思う	7
どちらでもない	1
少し思わない	3
思わない	0

表 4 「LogoLabo」は使いやすかったか

	人数
そう思う	0
少しそう思う	1
どちらでもない	5
少し思わない	5
思わない	2

表 9 各項目の平均

	表 3	表 4	表 5	表 6	表 7	表 8
平均	3.31	2.15	2.92	2.31	2.77	3.46

表 5 アボカドに沿ったフォントか

	人数
そう思う	1
少しそう思う	6
どちらでもない	1
少し思わない	2
思わない	2

表 6 既存のロゴと比べこのロゴを使いたい

	人数
そう思う	0
少しそう思う	2
どちらでもない	4
少し思わない	5
思わない	2

表 10 生成時間計測結果

評価者	「LogoLabo」の自動生成	評価者の手動生成
A	3分	7分
B	1分	14分
C	2分	21分
D	4分	8分
E	11分	26分

表 7 再度「LogoLabo」を使いたいと思ったか

	人数
そう思う	0
少しそう思う	5
どちらでもない	5
少し思わない	2
思わない	1

表 8 自分で生成する方法と比べ早く生成できたか

	人数
そう思う	4
少しそう思う	6
どちらでもない	1
少し思わない	1
思わない	1

でのロゴの生成時間に関しては最大 10 分の差があり、5 人全員が「LogoLabo」での生成がその他の生成方法と比べ時間が短くなる結果だった。アボカドに対するフォントのコサイン類似度は表 11 に示した。表 11 の通り、各フォントでのコサイン類似度はすべて高い値をとっており、差異が微量であった。

6. 考察

評価実験の結果より生成速度は人が生成する方法と比べ速い。しかし、既存のロゴと比べロゴとしての精度は不十分である。「LogoLabo」での生成時間と比べ手動での生成時間は平均 6.24 倍時間を要した。生成速度の項目が平均 3.46 とほかの項目と比べ高い値をとった理由は、アボカドとフォントの種類を限定することにより、事前に学習ができアプリケーション動作時には影響しないためである。「『LogoLabo』は使いやすいか」の項目では、評価が低く平均が 2.15 という結果である。原因としてアプリケーション自体の UI が見づらい側面もあるが、大半の原因はフォントに対する印象評価が長いことで画面が細かくなってしまい、さらに見づらくなるためである。「文字がアボカドに変化しているか」の項目では平均 3.31 であり、半数以上がアボカドに近い文字だと評価した。さらに、客観的に考察するためフォントを「LogoLabo」でアボカド化させた画像と、アボカドを含む 4 種類の果物画像で特徴量を取り比較した。特徴量は総当たりマッチング (Brute-Forcematcher) を python で実装し、比較対象は web 上の画像、アボカド、バナナ、ドラゴンフルーツ、リンゴで比較を行った。表 12 は図 15 の各画像に対して特徴量をとったものであり、数値が小さければ二つの画像の類似度が高いとされる。アボカド画像 1 とアボカド画像 2 の値が 78.38 に対し、生成画像とアボカド画像 1 の値が 84.43 でありその差は 6.05。アボカドとは異なる画像 3 種類の平均値が 114.78 であり、アボカド同士の画像との差は 36.4 であった。特徴量ではほかの果物に比べ生成された画像とアボカド画像での特徴量の値が 84.43 と小さく、この数値より生成画像はアボカドに近い画像であると考えられる。文字がアボカドに近く生成できた理由はアボカド自体の構造が単純であり、pix2pix

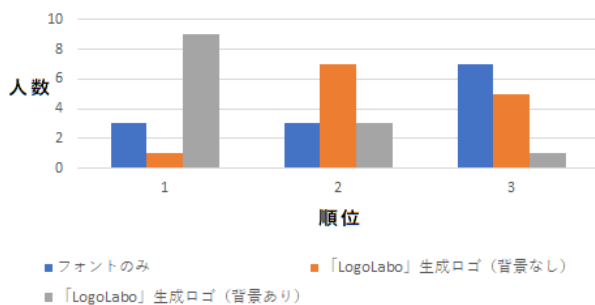


図 14 「LogoLabo」の生成画像の順位

が生成した背景がないロゴ (例:図 13b)、「LogoLabo」が生成した背景付きのロゴ (例:図 13c) について評価者のアンケートより順位付けを行ったものである。図 14 のグラフより最終的に生成される背景付き画像が、フォントのみとアボカドに変化された文字を含めた画像に対して、順位が高くなる傾向にあった。

「LogoLabo」でロゴを生成する方法と評価者がロゴを製作する方法でのロゴの生成時間は表 10 の通りである。なお、評価者 A・B・C には Intel(R) Core(TM) i9-10900k CPU と GeForce RTX 3080 GPU を搭載したパソコンで「LogoLabo」を動作させ、評価者 D・E には Intel(R) Core(TM) i7-8565U CPU が搭載されたパソコンで「LogoLabo」を動作させた。「LogoLabo」でのロゴ生成と評価者が手動でロゴを生成する場合どちらとも個人差があった。「LogoLabo」

表 11 コサイン類似度

フォント名	A	B	C	D	E
851 テガキカット	0.8238	0.8313	0.8649	0.9119	0.7605
font1	0.9790	0.8360	0.8915	0.9498	0.8127
源暎ゴシック P ExtraLight	0.9321	0.7550	0.9104	0.9127	0.7154
Nishiki-teki	0.9300	0.8557	0.9664	0.9544	0.6723
装甲明朝	0.9097	0.7452	0.8258	0.9646	0.6939

で学習が行い易いためであると考えられる。「アボカドに沿った文字か」の項目では平均 2.92 であったが、原因としてフォントのブレンド割合が 5 対 5 になり易いことが挙げられる。表 11 にある通りすべてのフォントで高い類似度が取れてしまい、アボカドに近いと判定されてしまったためである。故に、実際はアボカドに似ていないフォントも選択される場合があり、アンケート結果が「どちらでもない」に近い値になったと考えられる。

表 12 各画像対の特徴量

アボカド画像 1 とアボカド画像 2	78.38
生成画像とアボカド画像	84.43
生成画像とドラゴンフルーツ画像	117.95
生成画像とリンゴ画像	85.30
生成画像とバナナ画像	141.11

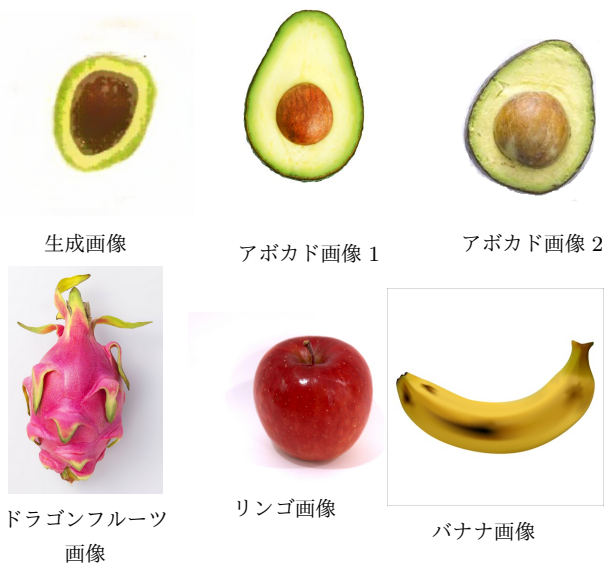


図 15 特徴点用画像

7. おわりに

本研究では、タイトルの文字列からフォントの生成と着色を行い、文字の形をした果物を含むタイトルロゴの自動生成の行える「LogoLabo」の実装を行った。「LogoLabo」は評価実験より、他のロゴ生成方法より早く生成でき、フォントをアボカドに近い画像に変化を行えた。しかし、アンケート項目が長く、「LogoLabo」は一度目だけアンケートを記入する必要があるため、今回の評価実験では、評価者は一

度だけのロゴ生成であったため、GUI の評価が低いという側面もあった。さらに、既存のロゴと比べ「LogoLabo」で生成したロゴ画像は、精度が低いことが判明した。

展望としては、まずアンケート画面の改善が必要である。解決案にアンケートの量を減らすことも考えられる。しかし、コサイン類似度で判定する方法ではアンケートの量が減ることによって、フォントの多様性が低くなるため、別の手法をとる必要がある。または、今回のフォントより特徴の差異が分かり易いフォントを使用することが効果的だと考えられる。次に、背景生成の部分はランダム生成であるので、客観的に美しいとされる黄金比を組み込むなど処理の開発を行っていく。

参考文献

- [1] 佐藤 翔, 石橋 柚香, 南谷 涼香, 奥田 麻友, 保志 育世, 吉田 光男: Twitter からの言及数が多い論文は言及されたことのない論文と比べてタイトルが「面白い」, 情報知識学会誌 18 巻 3 号 p. 146-150 (2020)
- [2] 佐川 友里香, 萩原 将文: 感性語を考慮した敵対的生成ネットワークを用いたロゴマーク生成, 日本感性工学会論文誌 18 巻 3 号 p.215-222 (2019)
- [3] Alexander Sage, Eirikur Agustsson, Radu Timofte, Luc Van Gool: Logo Synthesis and Manipulation with Clustered Generative Adversarial Networks, IEEE Conference on Computer Vision and Pattern Recognition, pp.5789-5888 (2018)
- [4] 山下 利之: 企業ロゴにおけるグローバル企業のローカル展開, 感性工学 18 巻 3 号 p.146-150 (2020)
- [5] 川島 拓也, 築館 多藍, 細谷 美月, 山浦 祐明, 山浦 祐明, 中村 聡史: 商品選択においてフォントがユーザの選択行動に及ぼす影響の調査, 電子情報通信学会技術研究報告, 119 号, p.113-118 (2019)
- [6] 濱野 花莉, 細谷 美月, 佐々木 美香子, 山浦 祐明, 中村 聡史: フォントと味の印象が商品選択行動に及ぼす影響, ヒューマンインタフェースシンポジウム論文集, No.5A-2, (2019)
- [7] 齊藤 絢基, 中村 聡史: 動的なフォント融合による文字デザイン支援手法, 研究報告エンタテインメントコンピューティング, Vol.2018-EC-47, Issue.2, pp.113-118 (2018)
- [8] 中村 充志, 瀧澤 生, 星 泰成, 綱島 秀樹, 陳 キョウ: 画像の感性を反映させたフォントの自動生成手法, 日本感性工学会論文誌, 17 巻 5 号 p.523-529 (2018)
- [9] zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks: <https://kaonashityc.github.io/2017/04/06/zi2zi.html> (2021.2.18 閲覧)
- [10] 鬼沢 武久, 矢幡 有朋: ユーザの持つ印象を反映した対話型ロゴデザインシステム, 日本知能情報ファジィ学会 ファジィシステムシンポジウム講演論文集 vol.32 pp.97-102 (2016)
- [11] Tai Vu, Robert Yang. GANime: Generating Anime and

- Manga Character Drawings from Sketches with Deep Learning (2020).
- [12] <https://en.m.fontke.com/font/18582253/> (2021.2.18 閲覧)
 - [13] <https://en.m.fontke.com/font/74126334/> (2021.2.18 閲覧)
 - [14] <https://okoneya.jp/font/download.html> (2021.2.18 閲覧)
 - [15] <https://fontdasu.com/966> (2020.12.30 閲覧)
 - [16] <http://flopdesign.com/blog/font/5228/> (2021.2.18 閲覧)
 - [17] <https://en.fontke.com/> (2021.2.18 閲覧)
 - [18] <https://fontdasu.com/> (2021.2.18 閲覧)
 - [19] <https://okoneya.jp/font/> (2021.2.18 閲覧)
 - [20] <https://www.liptonchilled.com/product/> (2021.2.18 閲覧)
 - [21] <https://www.liptonchilled.com/sp/product/1000/muscattea.html> (2021.2.18 閲覧)
 - [22] Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A:Image-to-Image Translation with Conditional Adversarial Networks, Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on(2017).
 - [23] <https://www.eatwholly.com/recipes/> (2021.2.18 閲覧)