

# 絵の具の混色を支援するシステム

土井遼太郎 岡部誠

静岡大学大学院 総合科学技術研究科 工学専攻 数理システム工学コース

E-mail: [ddd1150rrr@gmail.com](mailto:ddd1150rrr@gmail.com), [m.o@acm.org](mailto:m.o@acm.org)

## 1 概要

本研究の目的は、絵の具を混ぜて色を作るとき、誰でも迅速かつ正確に欲しい色を作れる世界を実現することである。デジタルアートの人気が高まる現代ではあるが、アナログアートの人気は衰えておらず、携わる人の人口は国内でも数百万人に上る([1,2])。しかし、アナログアートにおいて絵の具を混ぜて色を作るという基本的な作業(混色)が実は多くの人にとって難しく、欲しい色を思うように作れる人は少ない(図 1(a))。

そこで本研究では、今作りかけの色(混色のベースとなる色)と実際に作りたい色の 2 色を指定すると、絵の具(例えば一般的な 12 色セットの油絵の具)をどのように混ぜ合わせれば目的の色が作れるかを予測して教えてくれるシステムを開発する。システムの指示に従って色を混ぜ合わせることで、ユーザは目的の色を数分で作れるようになる(図 1(b))。

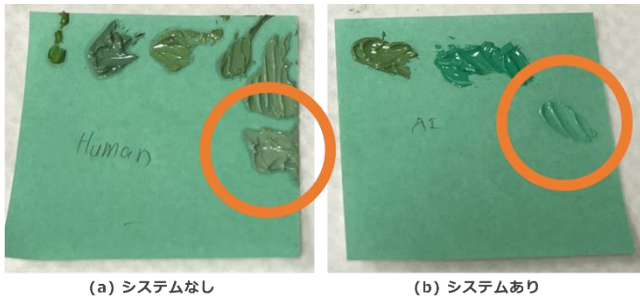


図 1: 色紙と同じ色を作ろうとした際に、システムを使用しなかった場合(a)と使用した場合(b)による結果の違い。オレンジ色の丸印で囲った色がそれぞれ最終的にできた色である。システムを利用せずに自力で混色を行った場合は、10分以上時間をかけて何度も混色を行っても色紙の色に近い色を再現することはできなかった。一方、システムを利用して何度か混色を行った場合は、6分ほどで色紙の色にかなり近い色の再現に成功した。

## 2 研究背景

アナログアートにおける基本的な作業として、絵の具同士を混ぜ合わせて色を作る「混色」という作業が存在する。混色による色表現はアナログアートにおける欠かせない技法の一つであり、作品の表現力を高める上で非常に重要である。絵の具を混ぜ合わせることによって、単色では表現しきれないような独自の色を作りだすことができるのである。

しかし、実際のところ混色という作業は多くの人にとって難しく、欲しい色を思い通りに作れる人は少ない。例えば、図 2(a)はゴッホの有名な作品「ひまわり」であるが、このひまわりの花びらの色を自分の絵で使ってみてみたいと思ったとしよう。ところが、実際に手元にある絵の具は図 2(b)のような 12 色のみであり、これらをどのような比率で混ぜればこの色になるかを瞬時に分かる人はいない。色作りが得意な人であれば、ある程度時間をかければこの色を再現することができるかもしれないが、たとえプロの画家であっても試行錯誤を繰り返す必要があり、即座に作れるわけではない。実際に、図 2(c)は色紙の色を絵の具で再現することを試みた際の結果の一例であり、どの色の絵の具をどのくらいの比率で混ぜ合わせる必要があるかを正しく判断することが非常に困難であることが分かる。

現在、絵の具の混色を支援するシステムやサービスはほとんど存在しない。混色を支援する数少ないシステムの一例として、カラーセンサで認識した物体の色を絵の具で自動生成する装置[4]が存在するが、こちらを使用するには専用のデバイスが必要になってしまう。

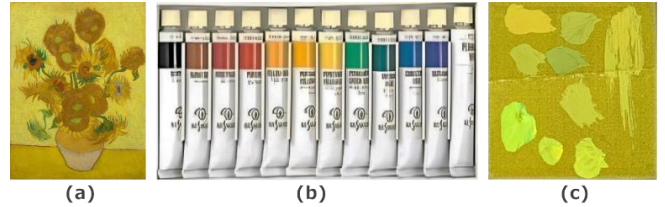


図 2: (a).ゴッホの最も有名な作品の一つである“ひまわり”。(b).一般的な 12 色セットの絵の具(クサカベ習作用油絵具 [3])。 (c).色紙と同じ色を絵の具で再現することを試みた結果の一例。何度もチャレンジしたが色紙と同じ黄色を作りことはできなかった。

## 3 提案システム

提案システムでは、混色のベースとなる色に対して、どの色の絵の具をどのくらい混ぜれば目標の色になるかを推測する学習モデルを構築して活用する。実際にシステムを利用する際の流れを図 3 に示す。スマートフォンのカメラアプリやインターネット上の画像などから取得した作りたい色の RGB 値と、混色のベースとなる色の RGB 値を提案システムに与えることによって、あらかじめ用意された絵の具(図 2(b))の中の、どの色をどのくらいの比率でベースの色と一緒に混

れば目的の色を再現できるかを確認することが可能となる。また、混ぜる比率を間違えてしまった場合や、一度の混色では思い通りの色にならなかった場合においても、その状態の色を混色のベースとなる色として再びシステムに与えることによって、その色に対してどの色をどのくらい追加すれば良いかということも確認することができる。

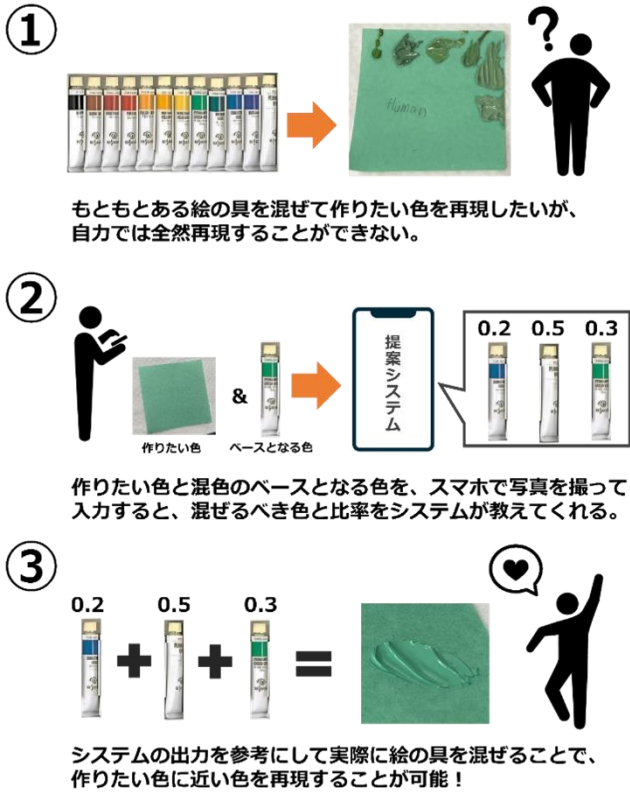


図 3: 提案システムを利用する際の流れ。作りたい色と混色のベースとなる色の 2 つの色の RGB 値を入力することで、どの色をどのくらい混ぜるべきかを合計が 1 となるような 0.1 刻みの数値で出力してユーザの絵の具の混色を支援する。

### 3.1 ニューラルネットワーク構造

提案システムはニューラルネットワークモデル(図 4)を作成することで実現する。ニューラルネットワークは Keras の functional API モデル[5]を用いて実装する。入力を作りたい色  $C_{target}$  と、ベースとなる色  $C_{base}$  である。  $C_{target}$  と  $C_{base}$  は RGB 値であるため 3 次元ベクトルとなり、入力は合計 6 チャンネルとなる。出力はあらかじめ設定された 12 色の絵の具の色とベースとなる色のそれぞれの予測混合比率に対応した値  $r_i$  であり、  $i$  は  $1 \sim N+1$  である。  $N$  は使える絵の具の数であり、今回は図 2(b)で示した絵の具セットの絵の具を用いるため  $N=12$  となり、出力の合計は 13 チャンネルとなる。  $r_1 \sim r_N$  はあらかじめ用意された絵の具のそれぞれの色の混合比率、  $r_{N+1}$  はベースの色の混合比率であり、  $\sum r_i = 1$  である。中間層は 4 つの全結合層からなっており、それぞれの層の活性化関数として ReLU 関数を適用する。また、それぞれ

の全結合層に Batch Normalization を一層ずつ適用し、出力層の前に Dropout を一層適用することによって、ニューラルネットワークの精度向上を図る[6]。出力層では活性化関数に Softmax 関数を適用することによって、どの色の絵の具をどれだけ混ぜるかを、  $\sum r_i = 1$  となる  $0 \sim 1$  の確率値に変更して予測結果を出力する。  $r_i$  が大きい色ほど混ぜる比率が大きくなることを意味する。

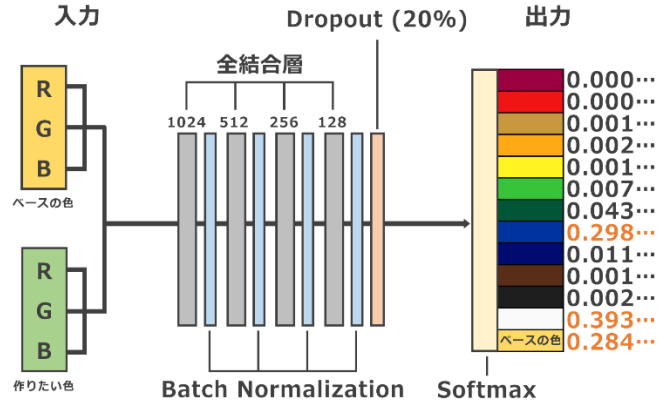


図 4: 使用するニューラルネットワークモデルの構造の概要図。この図の例では、黄色(混色のベースとなる色)の RGB 値と、緑色(作りたい色)の RGB 値が入力となる。この図における出力の例では、青色と白色、そして混色のベースとなる色の数値が他と比べて高いことから、これら 3 つの色を混ぜると良いということを意味している。

### 3.2 データセットの作成

ニューラルネットワークモデルの学習には、様々な色と比率の絵の具の混合によってどんな色ができるかというデータセットが大量に必要となる。しかし、データセットを作成するにあたって、実際の絵の具を混ぜていては膨大な時間がかかってしまう。デジタルペイントソフトを利用するという方法も考えられるが、デジタルペイントソフト上では基本的に実際の絵の具と色の混ざり方が異なる。そこで、本提案では、リアルな絵の具の混色結果を再現することが可能な MixBox というツール[7]を応用してデータセットを作成する。

ニューラルネットワークを学習するためのデータセットにおける 1 つのセットは、ベースの色  $C_{target}$ 、作りたい色  $C_{base}$ 、比率集合  $\{r_i\}$  から成る。まず、ベースの色  $C_{base}$  については、ランダムな 3 次元ベクトルを生成して決める。次に、比率集合  $\{r_i\}$  についてもランダムに決定するが、今回は  $r_1 \sim r_N$  のうち 2 つのみが正の値を持ち、他の  $N-2$  個については  $r_i = 0$  とする。また、ベースの色の比率  $r_{N+1}$  の値もランダムに決める。ただし、正の値をもつ  $r_i$  は  $0.1 \sim 0.9$  までの 0.1 刻みの数値であり、  $\sum r_i = 1$  である。最後に MixBox によって、作りたい色  $C_{target}$  が算出される。  $N$  色あるうちの 2 色のみを使う理由は、ベースの色に対して追加する色の数が 3 色以上になってしまうと実際に行う混色の作業が煩雑になってしまう、システ

ムのユーザビリティが低下すると考えたためである。

### 3.3 端数処理と結果の可視化

モデルに出力される比率集合  $\{r_i\}$  は、 $\sum r_i = 1$  となるような  $0 \sim 1$  の確率値として出力されるが、それらの値は浮動小数点数の値である。細かな値を示したところで、人間は混ぜ合わせる絵の具の量をそこまで正確にコントロールすることはできない。そのため、 $r_i$  の値が  $0$  から  $0.9$  までの  $0.1$  刻みの数値になるように、Python の標準ライブラリ `Decimal` における `quantize` メソッドを使用して、小数点第二位を四捨五入する端数処理を行う。また、一度の混色でベースとなる色に対して追加する色の数が多過ぎてしまうと、ユーザがシステムの予測通りに混色を行うことが困難になってしまう。そのため、ベースの色の混合比率  $r_{N+1}$  以外の  $r_i$  において、値が最も大きい  $2$  つを  $r_a$  と  $r_b$  として残し、それ以外の  $r_i$  は値を  $0$  とする。こうすることで、ベースとなる色に対して追加する色が  $2$  色以下になるような処理を行う。これらの処理によって  $\sum r_i = 1$  とならない場合は、 $\sum r_i = 1$  となるように  $r_{N+1}$  を増減させることで、図 5 の円グラフのように各出力を  $0.1$  刻みの数値で出力できるようにする。以上の処理と図 5 のような予測結果の出力方式によって、ユーザが実際に絵の具の混色を行う際にシステムの予測を参考にしやすくなる。

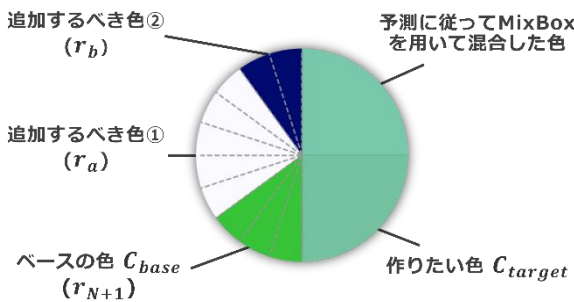


図 5: 円グラフによって予測結果を出力する際の例。円グラフの左側の色は、モデルが予測した作りたい色を作るために混ぜるべき色を示しており、その面積比はモデルが予測した混合比率に対応している。円グラフ右下の色はユーザによって入力された作りたい色  $C_{target}$  を示す。円グラフ右上の色は、MixBox を用いてモデルの予測に従った色と比率で混色した場合の色を示す。実際にはこの円グラフと共に各色の混合比率の数値も出力される。

## 4 結果と考察

提案システムは Python によるソースコードを用いて構築されており、Anaconda Navigator の Jupyter Notebook によってシステムを実行する。モデルの学習及び実験に使用した計算機の動作スペックは、CPU が Intel Core i7-7700 3.60GHz、メモリが 64GB、GPU が GeForce GTX 1080 Ti である。実際の絵の具の混色には、図 2(b)で示したクサカベ

習作用油絵具 12 色セットを用いる。学習における損失関数は Mean Squared Error を使用し、最適化アルゴリズムは Adam を使用した。また、バッチサイズは 2000、エポック数を 150 として、5000 万セットのデータセットで約 2 日間学習させた。また、実験は 3 波長系昼白色の照明環境下で行い、身の回りのモノから RGB 値を検出する手段として、「色スカウター」という iOS アプリケーション[8]を使用した。アプリケーションから取得した RGB 値が実際の色と異なる場合など、必要に応じて修正した後に、作りたい色  $C_{target}$  やベースの色  $C_{base}$  としてシステムに与えた。また、実際の絵の具の混ぜ方と MixBox での混ぜ方を比較した際に、実際の絵の具の方が黒の影響が強く、白の影響が弱いことが確認できた。そのため、システムの出力を参考にして実際に絵の具を混ぜる場合、白は出力の倍程度に、黒は出力の半分程度に混ぜることとした。

### 4.1 システムの実証実験

提案システムの指示通りに実際の絵の具を混色していくことによって、作りたい色を作ることができるかを確認する実証実験を行った。図 6 の左側の 5 つの折り紙の色を提案システムで作りたい色として与え、提案システムの混色予測を参考にして実際に絵の具を混ぜて色作りを行った。それぞれの結果を図 6 の右側に示す。それぞれの色を作るために提案システムによる混色予測を行った回数は、上から順に 3 回、2 回、4 回、3 回、5 回であった。作る色によって使用回数に差はあるが、提案システムを繰り返し使用し、混色予測を参考にして絵の具の混色を何度か繰り返し行うことによって、作りたい色に近い色を作れているということが確認できる。また、図 1 で示したように、提案システムを利用しなかった場合は 10 分以上時間をかけても目的の色を作ることができなかったが、提案システムを利用した場合は 6 分ほどで目的の色にかなり近い色の再現に成功した。なお、システムを利用しなかった際の混色回数は 6 回、利用した際は 3 回であった。これらのことから、提案システムの有用性を確認できた。

### 4.2 MixBox と実際の絵の具の混ぜ方について

システムを利用しても一度の混色で思い通りの色を再現するのは難しい。これは、MixBox での色の混ぜ方と実際の絵の具の混ぜ方が完全には一致しないからである。しかし、システムを利用して何度か混色を行うことで、図 6 のように目的の色を再現することができた。この理由としては、MixBox を使って色の混ぜ方を学習することで、システムは一般的な色の混ぜ方のパターンを学び、このパターンが実際の絵の具の混ぜ方にもある程度適用可能であったためであると考えられる。特に一度で作りたい色を再現できなくても、混色を何度か試すことで目指す色に徐々に近づいていくことが可能なシステムを実現できた。

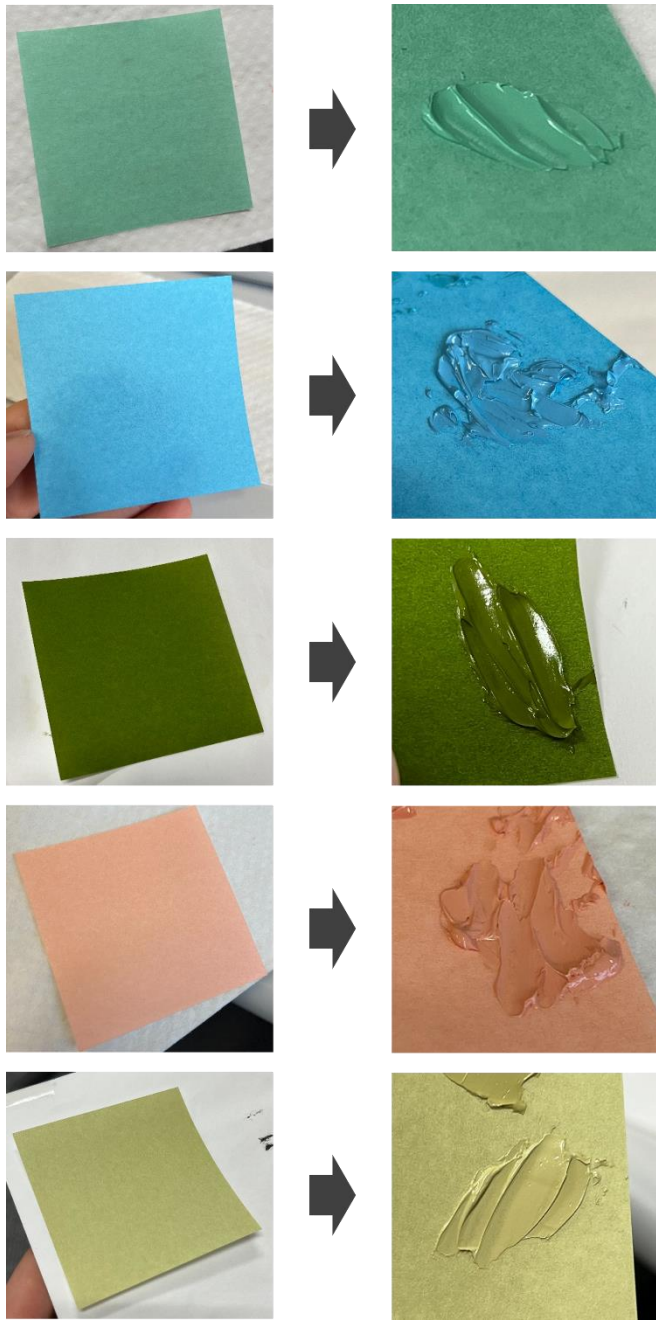


図 6: 作りたい色として提案システムに与えた色と、提案システムを参考に実際に絵の具の混色をした結果の例。

#### 4.3 失敗例

一方で、提案システムにおける混色予測が不適切であるケースや、提案システムを参考にしても作りたい色を上手に再現できないケース(図 7)なども存在した。そのため、今後はニューラルネットワークの精度を向上させることや、ファインチューニングなどによって実際に使用する絵の具の混ざり方を考慮した調整、使用する絵の具の組み合わせの調整が必要であると考えられる。

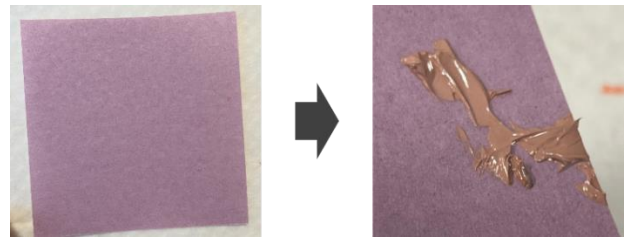


図 7: 提案システムを参考にして何度か混色を行っても作りたい色の再現が上手くいかない場合の一例。

## 5 まとめと今後の展望

本研究では、異なる色の絵の具同士の混ざり方をニューラルネットワークモデルに学習させることによって、作りたい色を再現するのに必要な絵の具の色と比率をモデルが予測し、絵の具の混色を支援するというシステムを提案した。

ユーザは作りたい色とベースとなる色の 2 つの RGB 値をシステムに与えることによって、あらかじめ設定された 12 色の絵の具の中からどの色をどれだけベースの色に対して混ぜるべきかを、提案システムによる予測から確認することができる。提案システムを使用した場合における、作りたい色の再現度や再現にかかった時間から、本システムによる絵の具の混色支援の有用性を示した。

今後はニューラルネットワークの精度を向上させることや、データセット及びニューラルネットワーク作成の際に実際に使用する絵の具の混ざり方を考慮することによって、提案システムにおける混色予測が不適切であるケースや提案システムを参考にしても作りたい色を上手に再現できないケースの発生を防ぐことを目標とする。また、スマートフォン上で使用できるようにアプリケーション化することや、あらかじめ設定した絵の具以外でのシステムの運用方法、RGB 値を適切に取得する方法などを検討することによって、より有用性の高いシステムを目指したいと考えている。

## 参考文献

- [1] 文化庁, 文化芸術関連データ集, 37. 我が国の「芸術家」人口①(職業別, 年齢別)。
- [2] 総務省統計局, 令和 3 年社会生活基本調査, 生活行動に関する結果, 趣味・娯楽, “男女, ふだんの健康状態, 頻度, 年齢, 趣味・娯楽の種類別行動者数(10 歳以上) - 全国”。
- [3] 株式会社クサカベ, 習作用油絵の具セット, [https://www.kusakabe-enogu.co.jp/product/cat\\_series\\_product?sid=101017#597](https://www.kusakabe-enogu.co.jp/product/cat_series_product?sid=101017#597)。
- [4] livedoorNEWS “見たままの色を絵の具で再現できるリアル「スポイトツール」が登場”, <https://news.livedoor.com/article/detail/14175853/>, 2023 年 6 月 6 日最終閲覧。
- [5] fchollet, Keras, “The Functional API”, [https://keras.io/guides/functional\\_api/](https://keras.io/guides/functional_api/)。

- [6] Xiang Li, Shuo Chen, Xiaolin Hu, Jian Yang, “Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2682-2690, 2019.
- [7] Šárka Sochorová, Ondřej Jamriška, “Practical pigment mixing for digital painting”, ACM Trans. Graph., vol. 40, Issue 6, pp. 1–11, 2021.
- [8] Hiroaki SAWANO, “色スカウター”, <https://apps.apple.com/jp/app/%E8%89%B2%E3%82%B9%E3%82%AB%E3%82%A6%E3%82%BF%E3%83%BC/id809465228>, 2023年6月8日最終閲覧.
- [9] 品田登”Kubelka--Munk理論とその応用”, 色材協会誌, 42(1969).