令和 6 年度

佐賀大学 理工学部 理工学科

知能情報システム工学コース

卒業論文

Google Street View による
空き家外観デザイン変更支援システム

提出者　：　荒木 大誠（21238010）

指導教員：　山口 暢彦　准教授

提出日　：　令和 7 年 2 月 3 日

Computer Science and Intelligent Systems Course,

Department of Science and Engineering,

Faculty of Science and Engineering, Saga University

A Graduation Thesis

Support system for changing the exterior design of vacant houses by Google Street View

Author	21238010	Taisei Araki
Supervisor	Associate Professor	Nobuhiko Yamaguchi
Submission	February 3rd, 2025	

概 要

建物外観のデザイン変更においては，外装のデザイン変更が地域の景観や文化に与える影響を正確に把握することが重要であり，外装の改修にはシミュレーションが非常に重要である．従来手法には，外観カラーシミュレーションアプリや 3D モデリングが提案されている．しかし，リアルな質感や色味を再現するのが困難で，専門的な技術が必要であるという問題点がある．そこで本研究では，簡便かつ低コストで現実味のある質感や色味をもったデザインでのシミュレーションを実現できる Google Street View と GAN（Generative Adversarial Network）を用いた空き家外観デザイン変更支援システムを提案する．本論文では，Google Street View から取得したヨーロッパの街並みと日本の空き家の画像を使って実験を行い，日本の空き家の外観デザイン変更の一例としてヨーロッパのレンガ風の建物に改修できることを確認した．

Abstract

Simulation is very important for exterior design changes because it is important to accurately understand the impact of exterior design changes on the local landscape and culture, and simulation is crucial for exterior renovations. Exterior color simulation applications and 3D modeling have been proposed as conventional methods. However, they are difficult to reproduce realistic textures and colors and require specialized techniques. In this study, we propose an exterior redesign support system for vacant houses using Google Street View and GAN (Generative Adversarial Network), which is a simple and low-cost way to simulate designs with realistic textures and colors. In this paper, we conducted experiments using images of European cityscapes and Japanese vacant houses obtained from Google Street View and confirmed that the system can modify the exterior design of a Japanese vacant house to a European brick-style building as an example of exterior redesign.

目次

図目次

表目次

第１章　はじめに

1.1 研究の目的と背景

近年日本における空き家の増加は深刻な社会問題となっており，総務省の「住宅・土地統計調査」によれば，図 1.1 で示している通り，2023 年時点で日本の空き家数は約 900 万戸に達し，空き家率は住宅総数に対して過去最高の 13.8%となった．この空き家の増加は、単に住居の問題に留まらず，老朽化が進行することによる景観の劣化，防犯上のリスク，災害時の危険性など，多岐にわたる社会的課題を引き起こしている．特に，長期にわたって空き家が放置されることにより，地域の住民や周辺環境への影響が深刻化しており，再活用することが求められている[1]．

一方で近年，外国人による日本の建築様式への関心が高まっており古民家の空き家の改修も広まっている[2]．しかし古民家のリノベーションにおいては，外装のデザイン変更が地域の景観や文化に与える影響を正確に把握することが重要であり，外装の改修にはシミュレーションが非常に重要である[3]．従来手法には，外観カラーシミュレーションアプリや 3D モデリングが提案されている．しかし，従来手法の外観カラーシミュレーションアプリでは，比較的簡単にデザイン変更のシミュレーションが可能であるが，リアルな質感や色味を再現するのが困難であるという問題点がある[4]．また，3D モデリングは複雑な操作が必要であり，専門的な技術が必要であり初心者には難しいという問題点がある[5]．その問題点に対して，簡便かつ低コストで現実味のある質感や色味をもったデザインでのシミュレーションを実現し，外国人をはじめとする空き家改修を行う人を支援して日本の空き家問題対策を促進する．

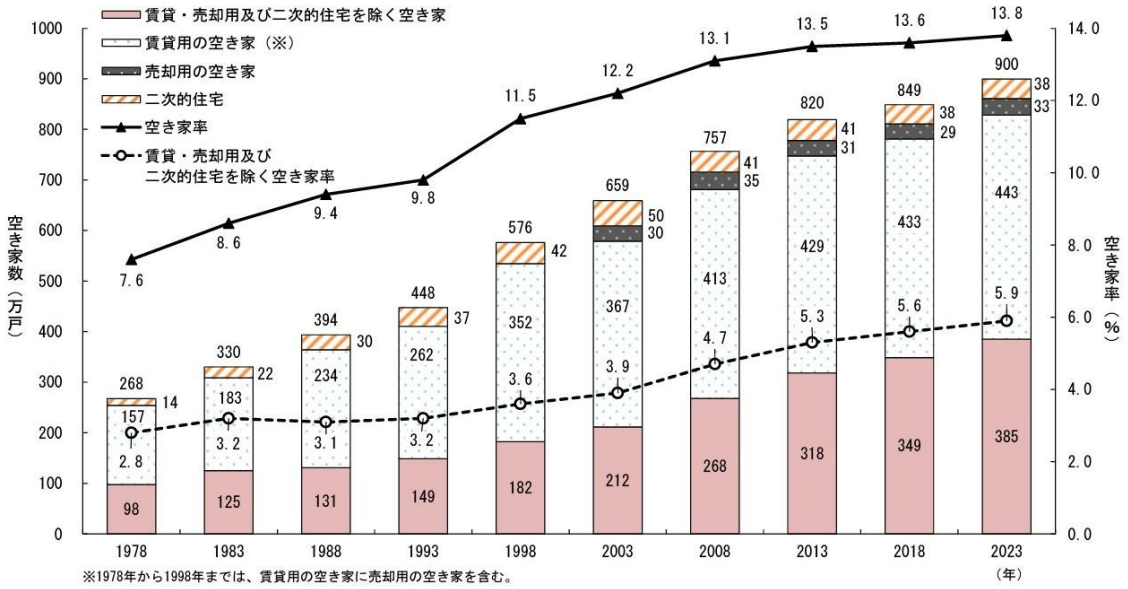


図 1.1　令和５年住宅・土地統計調査　住宅及び世帯に関する基本集計[1]

そのために，本研究では GAN（Generative Adversarial Network）を活用する建物デザイン変更支援システムを提案する． GAN を用いた学習によって様々なデザインに適用でき，より広範囲の画像生成が可能である．関連研究[6]でも，GAN を導入したデザイン生成手法を提案しており，和風な街並みとして，「京都「産寧坂」」，西洋的な街並みとして「スコットランド「エディンバラ」」を選定し，画像 200 枚に水増しを加えた，計 400 枚の画像を GAN で学習している．しかし，GAN の学習でより精度の高い結果を得るには，莫大なデータセットの数が必要であり，手作業での画像収集は大変困難である．そこで本研究は，画像収集において，Google Street View を活用し，自動化を実現する．Google Street View から自動で画像を収集するため，海外のような画像収集に手間と時間がかかる地域の画像であっても，短時間で大量のデータを簡単に集めることができる．本論文では，Google Street View から取得したヨーロッパの街並みと日本の空き家の画像を使って実験を行い，日本の空き家の外観リノベーションの一例としてヨーロッパのレンガ風の建物に改修できることを確認した．

1.2 論文構成

本論文では，Google Street View と GAN を用いた建物デザイン生成システムを提案する．セマンティックセグメンテーションはアノテーションをして作成したデータセットを YOLO(You Only Look One)で学習させることにより実装する．そこで得た画像内のセグメントした建物のみを Python により特定の色で塗りつぶし，マスク画像を生成する．その結果得た街中の建物のマスク画像と元画像をペアで対応させて学習させる．そこで学習させた Generator を用いて日本の空き家に対してマスク画像を作成し，新たなデザインを生成する．本論文の構成は以下の通りである．2 章では，画像収集で使用する Google Maps API について，3 章は，YOLO とセグメンテーションについて，4 章では研究で使用した GAN について，第 5 章では本研究で行った実験について述べる．

第 2 章　Google Maps API

ここでは Google Maps API と実際に使用した結果について述べる．

2.1 Google Street View について

Google Street View とは，対象地域全体について，指定された道路からの 360 度のパノラマ画像を提供する Google のサービスのことであり，道路や店内，施設内といった屋内版の 360 度パノラマ画像を閲覧可能である．2007 年にサンフランシスコ，ニューヨーク，ラスベガス，マイアミ，デンバーの街並みから開始し，現在は世界の 100 以上の国や地域をカバーしている．日本のストリートビューは 2008 年から始まっている．

2.2　API について

API とは「アプリケーション・プログラミング・インタフェース（Application Programming Interface）」の略称で，ソフトウェアやプログラム，Web サービスの間をつなぐインターフェースのことである．インターフェースというのは「境界面」や「接点」の意味を持ち異なる 2 つものをつなぐという意味である．

2.3　Google Maps API の概要

Google Maps API とは Google Street View が提供するアプリケーション・プログラミング・インタフェースである．Google Street View API の対象地域は，Google マップアプリケーションの対象地域と同じである．Google Maps API には，主にマップ(Maps)，ルート(Routes)，プレイス(Place)，環境(Environment)の 4 つの種類に分かれている．以下の表 2.1 に 4 つの種類と詳細をまとめる．また，表 2.2 は Geocoding API の 2 つの機能と説明をまとめている．

表 2.1　Google Maps API の種類と説明

種類	説明
マップ(Maps)	Google マップでの地図表示やストリートビューなどの機能を利用するための機能
ルート(Routes)	渋滞情報を考慮したルート検索や，指定した地点間のルート検索，最適巡回機能を利用するための機能
プレイス(Places)	日本国内を含む，全世界 1 億以上のスポット情報の掲載と，施設名や住所，電話番号による検索を利用するための機能
環境(Environment)	サステナビリティに関連するサービスとして，大気質や太陽光等に関する最新のデータ検索機能を利用するための機能

表 2.2　Geocoding API

機能	説明
ジオコーディング	「1600 Amphitheatre Parkway,Mountain View, CA」のような住所を緯度と経度の座標またはプレイス ID に変換する．これらの座標を使用して，地図上にマーカーを配置したり，ビューフレーム内で地図の中心や位置を変更したりできる
リバースジオコーディング	緯度と経度の座標またはプレイス ID を人が読める住所に変換する．住所は，配達や集荷など，さまざまなシナリオで使用できる

第 3 章　YOLOv8

ここでは YOLOv8 と実際に学習させた結果について述べる．YOLO は YOLO(You Only Look Once）の一種である．YOLO は 2015 年にワシントン大学のジョセフ・レッドモン（Joseph Redmon）とアリ・ファルハディ（Ali Farhadi）によって開発された，一般的な物体検出と画像分割モデルである．YOLOv8 は，リアルタイム物体検出器 YOLO シリーズの最新版で 2023 年 1 月に公開された．旧バージョンに比べ，精度とスピードの面で最先端の性能を提供する．YOLOv8 は，これまでの YOLO バージョンの進化をベースに，新機能と最適化を導入し，幅広いアプリケーションにおける様々な物体検出タスクに適用できる．次節で，YOLOv8 の概要から述べていく．

3.1 YOLOv8 の概要

かつての物体検出手法は，スライディングウィンドウ法や領域提案法などのアルゴリズムがあった．しかし，これらの手法は，まず画像のなかで物体を検出し，次に物体を個々に識別し詳細を確認するプロセスであった．ひとつひとつの領域に対して物体検出と識別を繰り返すため，計算処理が重い，時間がかかるというような課題があった．一方で，YOLO は画像を一度走査するだけで, 画像上の全ての物体を同時に検出し, 識別できる. そのため、計算処理の負担が大幅に軽減され，処理時間も短縮される．YOLOv8 は，YOLOv5 の公開元である Ultralytics 社が公開した YOLO の最新バージョンのモデルである．大規模データセットでの学習や CPU, GPU を始めとしたさまざまなハードウェアでの実行が可能になっている．

3.2　セグメンテーション

セグメンテーションは,画像処理やパターン認識などの分野で広く使用されている概念で, 画像を複数オブジェクトに分割することである.セグメンテーションにはセマンティックセグメンテーションとインスタンスセグメンテーションなどがあり，本研究の YOLOv8 ではセマンティックセグメンテーションを使って同じラベルの個体ごとの認識を行っている．

3.2.1　セマンティックセグメンテーション

セマンティックセグメンテーションは,画像の各ピクセルが所属しているクラスを特定する手法のことである．これにより，不定形の領域を高い精度で検出することができる．しかし, 物体の種類ごとに領域分割を行っているため, 物体が重なっているときにはそれぞれの物体を区別できないというデメリットもある.

セマンティックセグメンテーションは SegNet や FXN，U-NET などのさまざまなネットワークアーキテクチャが存在し，これらを使用することで医療画像診断や自動運転など

の技術に使用されている.

3.2.2 インスタンスセグメンテーション

インスタンスセグメンテーションは, 画像内の各物体の領域を特定した後, 個体ごとに領域分割とピクセル単位でセグメンテーションを行う. これにより, セマンティックセグメンテーションが苦手とする隣接した物体を区別でき, 画像内の物体の個々の位置や形状についての情報を取得することができる.

インスタンスセグメンテーションは R-FCN や FCIS, MASK R-CNN などのさまざまなネットワークアーキテクチャが存在する.

3.2.3 YOLOv8 の仕組み

YOLOv8 は End-to-end という手法が採用されている.

End-to-end は検出と識別を一貫してディープニューラルネットワークが行う方法で画像を入力するだけで物体検出と識別を行う. これにより学習プロセスが効率的になり, 一度の学習で物体検出の全工程を最適化できる.

3.3 Cvat

Cvat は Intel が開発しているオープンソースソフトウェアで画像や動画の自動アノテーションツールである. Cvat は Web アプリケーションとして提供されており, 主に React などで構成されており, ブラウザを介してアクセスし, インターネット上で動作する. 特徴は, 矩形（バウンディングボックス), ポリゴン, ライン, ポイント, セグメンテーションマスクなどさまざまなアノテーション形式をサポートしていることや, ディープラーニングモデルによる事前学習されたラベルの提案機能により自動的にアノテーションを予測する機能などがある.

3.3.1 アノテーション方法

まず, 取得した RGB 画像のアノテーションを 3.3 で説明した Cvat を用いて行う. 画像内に複数の建物がある場合でも同一のグループ ID を付与し, 区別は行わず, すべての建物をアノテーションする. このようにしてアノテーションを行い, 得たデータを Cvat から COCO 形式にエクスポートし YOLO 形式に変更する. この方法で, Google Street View により収集した東京都の画像 110 枚をアノテーションする.

3.3.2 アノテーション結果

元画像とこの方法で得たデータセットの画像をそれぞれ図 3.1, 3.2 に示す.

図 3.1　元画像

図 3.2　アノテーション結果

第 4 章　GAN

GAN は「敵対的生成ネットワーク」(Generative Adversarial Networks)のことであり，2014 年に Goodfellow らによって発表された画像生成モデルである．データセットから特徴を学習することで，新たなデータを生成したり，存在するデータの特徴に従って変換でしたりできる．

4.1　GAN の概要

本節は，論文[7]を参考に記述した．

GAN は Generator(生成器)と Discriminator(識別器)から構成されている．学習対象の画像データセットに対して，生成器は可能な限り，正解画像に近い画像を生成する．識別器は生成器により生成された画像が正解かそうでないかを判別するという役割があり，生成器と識別器の Min-Max の目的関数で最適化が行われる，すなわち，生成器と識別器が互いに競い合いながら両者の学習が進む．画像生成モデルには，GAN の他に Variational Autoencoder(VAE)や Autoregressive Model(AR)がある．VAE はサンプリングコストが低いものの，近似分布の推定を行うため必ずしも実際の生成分布を一致するとは限らない．AR は精緻な生成が可能であるが，生成が再帰的であるため並列化が難しく時間がかかるという問題点がある．そのため生成コストや生成の精度に課題がある．一方で，GAN は VAE と同様に，データの潜在的な表現を行うため生成結果がコントロールしやすく，サンプリングコスト低い．さらに最適な条件下の場合，精緻なデータ生成が可能である．しかし，Min-Max 最適化を行うため，学習が安定しないことがあるので注意が必要である．

4.2　GAN の仕組み

本章では，真のデータ分布 $p_t(x)$，生成データの分布 $p_g(x)$と定義する．GAN の目的は，

$p_t(x)$に一致するような $p_g(x)$を得ることである．これを実現するために，GAN では生成器と識別器の二つのニューラルネットワークを用いる．生成器(G)は乱数 $z \sim p_z(z)$からデータ空間 $x = G(z)$への写像を行う．識別器(D)はデータ x が $p_t(x)$からサンプリングされたものであれば，確率 $p = D(x) \in [0, 1]$を付与し，$p_g(x)$からサンプリングされたものであれば，確率 1-p を付与する．以下の数式(1)において生成器と識別器は最適化を行う[7].

$$\min_G \max_D V(D,G) = E_{x\sim p_t(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log\left(1 - D\left(G(z)\right)\right)] \quad (1)$$

$E_{x\sim p_t(x)}[\log D(x)]$は真のデータセットからランダムに選択された x を入力とした識別器(Discriminator)の結果のスコアである．$E_{z\sim p_z(z)}[\log\left(1 - D\left(G(z)\right)\right)]$はランダムノイズデータ z から生成器(Generator)によって生成された画像を入力された Discriminator の結果のスコアである．Discriminator が画像に対して正しく正誤判定ができるようになるとこの式の値は増加し，Generator がより本物に近い偽物画像を生成できるようになるとこの式の値は減少する．

4.3 使用した GAN の種類

4.2 で説明した GAN を使用して画像生成を行う場合，モデルの訓練が生成結果の真偽判定のみを評価指標として行われるため，生成内容を制御することは困難である．本研究では，ヨーロッパのレンガのデザインの建物画像を生成することを目的としており，生成した建物のデザイン画像はヨーロッパのレンガのデザインになる必要がある．GAN で生成する画像を制御する手法の一つとして，conditional GAN(cGAN)が提案されている．cGAN では，疑似データを生成する際の入力データに出力結果を制御する情報を加えることで生成される画像データを制御する．即ち，式(1)の Generator と Discriminator の入力 D(x)と G(z)がラベル情報 y を加えた D(x,y)，G(z,y)となる．

本研究では，cGAN の中の Pix2Pix を使用して画像生成を行う．Pix2Pix は条件として画像を使用する．画像データの特徴を学習し，類似した画像データを生成できるモデルである．すなわち，学習させた画像セットと似ているものを作り出すことができる．Pix2pix では条件画像と正解画像をペアとした 2 組の画像から関係性を学習し，1 枚の画像を生成する．Pix2Pix の特徴の一つとして，Generator のネットワークに U-net を用いることが挙げられる[8]．U-net のアーキテクチャはエンコーダとデコーダの構造をもつ．エンコーダは，畳み込みを用いて画像の幅と高さを縮小していくことで，画像の重要な特徴のみを抽出する．一方で，デコーダは，画像の幅と高さを拡大していき，エンコーダによって生成された潜在ベクトルを元に，元画像に似た画像を再構築する．また，U-net では，すべての情報を低解像度に圧縮(ダウンサンプリング)するのではなく，skip connection を設けることで同じ特徴を直接利用し，より精緻な画像生成を可能にする[8]．skip connection は，ディープニュ

ーラルネットワークにおいて，途中の複数層を N 層分スキップして先の層へとつなげる迂回パスにより，離れた層間で順伝搬・逆伝搬を行えるようにする機構である．この接続によって，情報が途中で失われることなく，細かいディテールまで復元されやすい．

第 5 章　実験

本章の構成は以下の通りである．5.1 では、Google Street View を用いた街中の画像の収集方法について述べる．5.2 では，Yolov8 を用いたマスク画像作成方法について述べる．5.3 では，GAN を用いた学習について述べる．

5.1　Google Street View を用いた画像収集

本研究では，Google Maps API によって，Google Street View と連携し，緯度と経度の最大値，最小値で定めた長方形領域の範囲から自動的に画像収集を行う．

5.1.1　実行環境

本研究では，Google Maps API は Python で実行する．使用する API の種類は Street View Static API と Geocoding API である． Geocoding API は，表 2.3 のプレイス(Places)の一種であり，住所，緯度と経度の座標，またはプレイス ID として場所を受け入れるサービスである．表 2.4 で示した通り，ジオコーディングとジオコーディングリバースがある．本研究では，ジオコーディングを使用し，緯度と経度の座標から住所に変換する． Street View Static API は，表 2.3 のマップ(Maps)の一種であり，JavaScript を使用せずに，静的（非インタラクティブ）なストリートビュー パノラマやサムネイルをウェブページに埋め込むことができる．ビューポート(表示領域)は，標準の HTTP リクエストによって送信された URL パラメータで定義する．すなわち，URL にパラメータを組み込む．パラメータは緯度と経度，水平方向の視野角，北を 0 度とした時の方向が指定可能である．これらのパラメータを元に対象地域の静止画像が取得できる．この 2 つの API を組み合わせることで， Google Street View から自動で画像とその地点の住所情報を取得する．
本研究では水平方向の視野角は 120 度，北を 0 度としたときの角度は，0 度から 1 枚取得するごとに 90 度増加させる．これを 4 回繰り返し，前後左右の 4 方向画像を取得する．

5.1.2　画像収集範囲の定義

Google Maps API によって画像を収集する範囲の定義を説明する．画像収集範囲は，緯度と経度それぞれの最小値と最大値による長方形領域を定める．その中で，1 つの座標につき前後左右の 4 方向画像を取得した．4 方向画像により道路に面した建物の正面画像を取得

できる．以下の図 2.5 はチェコ共和国の長方形領域である．

図 5.1　画像取得範囲の一例

5.1.3　画像収集の結果

図 5.2 は 5.1.2 で説明した手法を用いて実際に Google Street View より集めた画像の一例である．主な関数のパラメータは表 5.1 にまとめている．

表 5.1　画像収集の主な関数のパラメータ

緯度	49.219052
	49.219582
経度	17.648000
	17.653348
水平方向の視野角(fov)	120°
北を 0° とした方向角(heading)	0° ， 90° ， 180° ， 270°

図 5.2 画像取得の一例(チェコ共和国　ズリーン州)

5.2　Yolov8 を用いたマスク画像生成

本研究では，5.1 で説明した手法を用いて画像収集したデータを YOLOv8 により街中の建物のマスク画像を生成する．

5.2.1　建物のセグメンテーション

まず，街中の建物を検出するセマンティックセグメンテーションモデルを作成する．3.3.1 で説明した手法で作成したデータセットを用いる．そのうち，訓練データに 90 枚，検証データに 20 枚を割り当てる．学習回数は 150 回とし，バッチサイズは 16，Optimizer は Adam と設定した．図 5.3 は学習したモデルを用いたテスト結果である．

図 5.3　元画像とセグメンテーション結果

5.2.2　マスク画像作成

5.2.1 で説明した手法で作成したヨーロッパの街中の建物をセグメントした画像に対し

てセグメント領域を黄色(B=0，G=255，R=255)でオーバーレイする．図 5.4 はその結果の一例である．

図 5.4　元画像とマスク結果

5.3　GAN を用いた学習

本節では，　5.2 で説明した手法で作成したデータセットの学習について説明する．学習には 4.3 で説明した Pix2Pix を使用する．

5.3.1　環境構築

本研究では実験の環境として，OS は Windows10，プログラミング言語は Python3.7.1 を用いた．これは，画像やグラフの表示などグラフィカルな部分のサポートが充実している Jupyter notebook に対応させるためである．

Python は近年注目されているプログラミング言語である.特徴として少ないコードで簡潔にプログラムを書けることや専門的なライブラリがたくさんあることがある．

5.3.2　GPU マシンの使用

GAN による機械学習には莫大な計算量を要することで知られている．そこで，本研究をスムーズに進めるために GPU マシンを使用する．今回使用する GPU マシンの OS は Windows 11 を使用する．Windows 11 に anaconda を用いて仮想環境の作成を行った．Tensorflow 及び Tensorflow-gpu は 2.6.0，Keras は 2.6.0 をインストールする．Tensorflow は Google が開発したオープンソースの機械学習フレームワークである．Keras はディープラーニングモデルを簡単かつ迅速に構築するための高レベル API の機械学習ライブラリである．

5.3.3　画像の前処理

GAN などによる機械学習では，より多くのデータセットを訓練させる方がより精度の高い結果が得られることが知られている．そこで，本研究では，Google Street View によって収集したオリジナル画像 14584 枚に対して，拡張処理を行う．本実験で用いる拡張処理には以下の 2 種類の処理により拡張する．回転処理によって，より現実世界を再現し，撮影角度のバリエーション増加が期待できる．また，コントラスト変化により，晴れ．曇りなどの天候や日差しの強度の違いを再現でき，データセットの多様性を増やすことができる．

1.　回転
画像を 10°と 350°に回転させる．

2.　コントラスト変化
画像の明るさを 0.8 倍，1.2 倍，1.4 倍にそれぞれ変化させる．

これらの拡張処理により，元の 14584 枚と合わせて合計 87504 枚にデータを拡張できる．同様に，マスク画像の 14584 枚に対しても回転処理を行う．マスク画像に対するコントラスト変化の画像に関しては，コントラスト処理を行った元画像に対して，5.2 で説明した手法を用いて再度マスク画像を作成したものとする．これは，マスク処理後の画像にコントラストを行うのを避けて，マスクの色は統一して学習させるためである．これにより，変化対象のマスク部分の多様性を制限し，Generator が建物以外のデザイン変更を行うのを防ぐ．このような拡張処理を行った画像も元画像とマスク画像を対応付けて合計 87504 ペアを作成する．図 5.5，図 5.6，図 5.7，図 5.8，図 5.9 はその一例である．

図 5.5　10°回転

図 5.6　350°回転

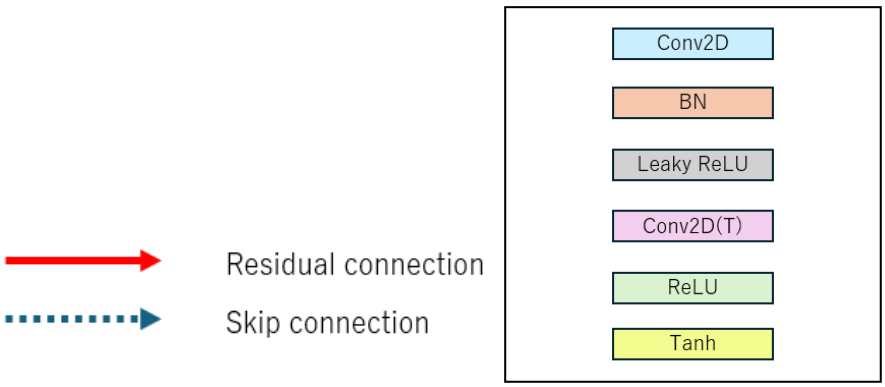
図 5.7　コントラスト 0.8 倍

図 5.8　コントラスト 1.2 倍

図 5.9　コントラスト 1.4 倍

5.3.4　生成器(Generator)

Generator のネットワークを図 5.10 に示す．本実験の Generator は，正解画像と条件画像を比べて異なる部分を学習し，条件画像を正解画像に近づけることを目的として画像生成する．主な構造は，4.2 で説明した U-net 構造である．本実験での Generator には，残差ブロックを利用する．残差ブロックは層を通過した後に入力をそのまま加算する手法である．U-net の skip connection において残差ブロックを組み込むことで，複雑で抽象的な特徴(高次元の特徴)を学習させるとともに，さらに深いネットワークでの勾配消失の軽減し，安定した学習を実現する．



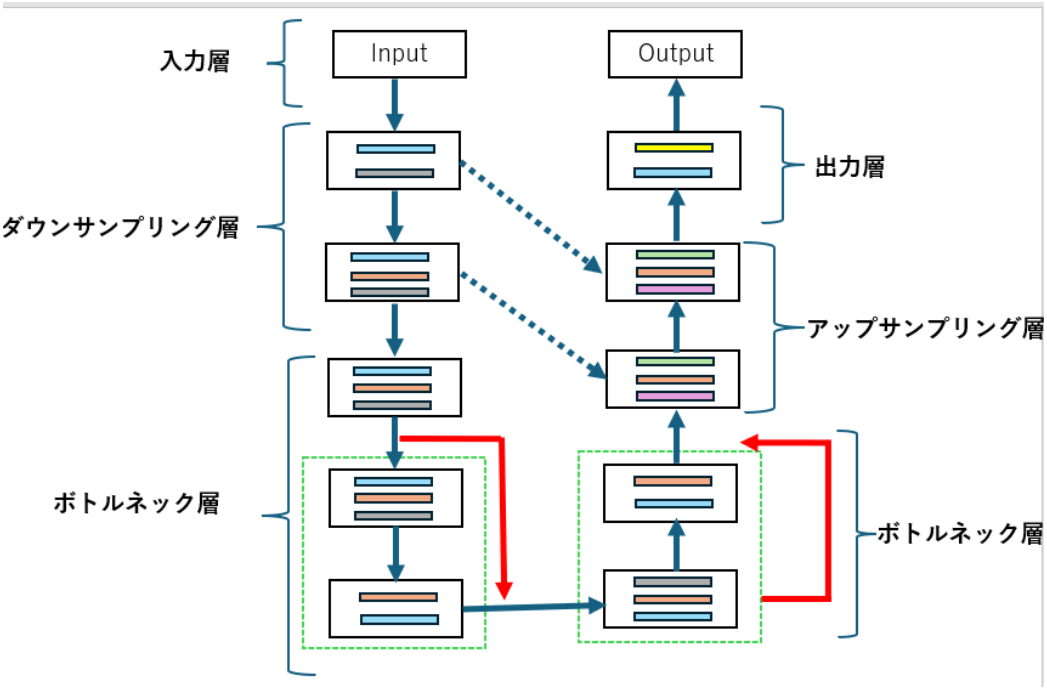


図 5.10　Generator のネットワーク

・入力層

入力は 256×256 の 3 チャネルの RGB 画像である．

・エンコーダ(ダウンサンプリング層)

層 1 : 64 個のフィルタ，4×4 カーネル，ストライドは 2，'same'パディングを使用した 2D 畳み込み層である．活性化関数 LeakyReLu の alpha 値は 0.2 とする．この層で入力画像の低次元の特徴(高解像度の特徴)を捉える．

層 2 : 128 個のフィルタ，4×4 カーネル，ストライドは 2，'same'パディングを使用した 2D 畳み込み層である．活性化関数 LeakyReLu の alpha 値は 0.2 とする．BatchNormalization 層を通して特徴量を正規化する．この層で入力画像の高次元の特徴(低解像度の特徴)を捉える．

・ボトルネック層(エンコーダ部最終層)

層 3 : 256 個のフィルタ，4×4 カーネルを使用した 2D 畳み込み層である．活性化関数 LeakyReLu の alpha 値は 0.2 とする．BatchNormalization 層を通して特徴量を正規化する．この層で残差ブロックを 2 回適用し，残差接続を行う．

・デコーダ(アップサンプリング層)

層 4 : 128 個のフィルタを持つ転置畳み込み層である．ストライドは 2，'same'パディングを使用する．活性化関数 ReLu と BatchNormalization を使用する．この層で，ダウンサン

プリング層の層 2 との skip connection によりダウンサンプリングで失われた低次元の特徴を再構築する．
層 5：128 個のフィルタを持つ転置畳み込み層である．活性化関数 ReLu と BatchNormalization を使用する．この層で，ダウンサンプリング層の層 1 との skip connection によりさらに低次元の特徴を再構築する．

・出力層
この層で3チャネルのRGB画像を生成する．カーネルサイズは4×4, ストライドは1, 'same' パディングを使用する．出力には，活性関数 tanh を適用する．

5.3.5　識別器(Discriminator)

Discriminator のネットワークを図 5.11 に示す．本実験での Discriminator は，正解画像と生成された画像の 2 つの入力を受け取り，生成された画像の真偽を示すスコアを出力する．主な構造は畳み込み層である．

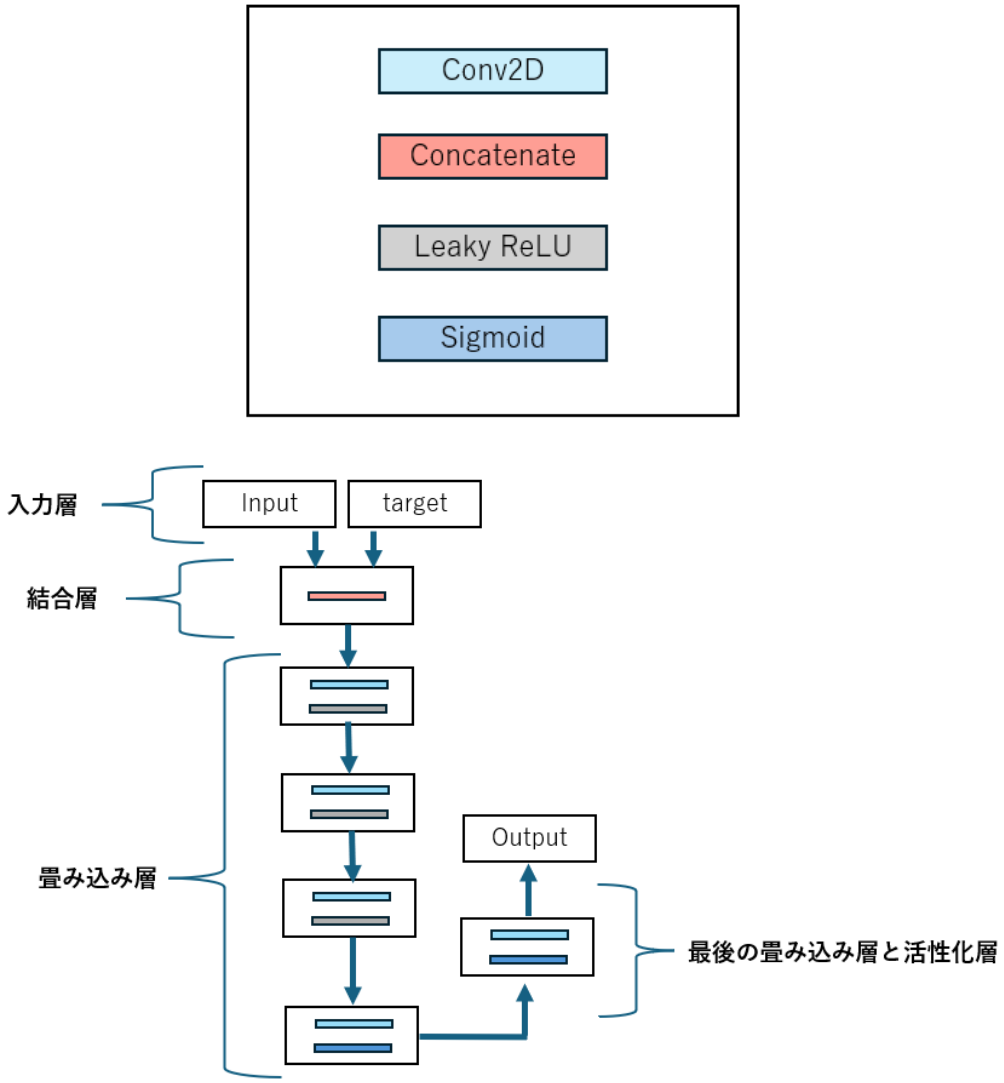


図 5.11　Discriminator のネットワーク

・入力層

実際の画像と生成された画像を受け取る．形状は 256×256 サイズの 3 チャネルである．

・結合層

実際の画像と生成された画像をチャネル次元で結合する．256×256 の 6 チャネルのテンソルが得られる．

・畳み込み層

層 1：64 個のフィルタ，4×4 のカーネル，ストライドは 2 で，’same’パディングを使用した 2D 畳み込み層である．

層 2：128 個のフィルタ，4×4 のカーネル，ストライドは 2 で，’same’パディングを使用し

た 2D 畳み込み層である
層 3 : 256 個のフィルタ，4×4 のカーネル，ストライドは 2 で，'same'パディングを使用した 2D 畳み込み層である
層 4 : 512 個のフィルタ，4×4 のカーネル，ストライドは 2 で，'same'パディングを使用した 2D 畳み込み層である

・最後の畳み込みと活性化層
1 個のフィルタ，4×4 のカーネル，活性化関数 Sigmoid を適用する．

5.3.6 損失関数

・L1 Loss（L1 損失）
生成画像と元画像とのピクセルごとの平均絶対誤差を測定する．画像全体のスムーズさや明瞭さを維持するために寄与する．以下の数式で表される． y_{true}を正解画像のピクセルの値，y_{pred}をモデルの予測のピクセルの値，N を全体のピクセル数とする．

$$\text{L1 Loss} = \frac{1}{N}\sum_{i=1}^{N}\left(\left|y_{\text{true}} - y_{\text{pred}}\right|\right) \tag{2}$$

・Perceptual Loss（知覚損失）
単純なピクセル単位の差異ではなく，VGG19 などの事前学習モデルを用いて正解画像と生成画像の特徴マップを抽出しその差分を使用する．出力する層の特徴を適切に選択することで，画像の高次元の特徴を維持するために寄与する．

・BCE Loss（バイナリークロスエントロピー損失）
Generator の損失：生成画像(偽画像)を識別器が本物を判定するように促す．
Discriminator の損失：正解画像を本物，偽画像を偽物と判定できるように促す．
GAN 特有の損失関数であり，生成画像のリアリティの向上に寄与する．実際のラベルと予測確率の間の距離を測定する． y を実際のラベルとする．0 または 1 をとる．$\hat{y}$をモデルの予測値とする．0 と 1 の間の確率を表す．以下の数式で表される，

$$\text{BCE Loss} = -\sum[y \times \log(\hat{y}) + (1-y) \times \log(1-\hat{y})] \tag{3}$$

Generator は上記の損失関数を加重平均して最適化を行う．以下の数式で表される．λ_{L1}は L1 損失の重み，$\lambda_{\text{perceptual}}$は知覚損失の重み，$\lambda_{\text{BCE}}$はバイナリークロスエントロピー損失の重みである．本研究では，λ_{L1}は 100，$\lambda_{\text{perceptual}}$は 25，$\lambda_{\text{BCE}}$は 1.5 に設定した．

$$\lambda_{\text{L1}} \times \text{L1 Loss} + \lambda_{\text{perceptual}} \times \text{Perceptual Loss} + \lambda_{\text{BCE}} \times \text{BCE Loss} \tag{4}$$

Discriminator は，以下の数式を最小化する．BCE(rea)は正解画像が本物と判別される損失で数式(3)において y を 1 としたもので，BCE(fake)は偽画像が偽物と判別される損失で数式(3)において y を 0 としたものである．

$$\mathrm{BCE(real)} + \mathrm{BCE(fake)} \tag{5}$$

5.3.7　学習結果と考察

5.2 や 5.3.3 で説明した手法で作成したデータセットを用いて Pix2Pix により学習を行う．画像サイズはすべて 256×256 で統一する．また，全 87504 ペアのうち，訓練データに全体の 80%を割り当て，検証データに全体の 20%を割り当てる．学習回数は 200 回とし，バッチサイズは 16 とした．学習時間は約 60 時間である．図 5.12 は入力画像，図 5.13 は出力画像の一例である．図 5.11 は元画像と出力画像の一例である．

図 5.12　入力画像

図 5.13　出力画像

図 5.14　元画像と生成画像 1 [9]
(元画像引用元： Shutterstock)

図 5.15　元画像と生成画像 2 [10]
(元画像引用元：不動産売却コラム 八城地建)

図 5.13 の結果のように，図 5.10 のような黄色のマスク画像を入力すると，ヨーロッパのレンガ風のデザインに変更した建物を生成することができた．図 5.13 から，元画像は一軒家に対して，生成画像はアパートメントのような建物に変換されていることが分かる．これは，学習させたヨーロッパのデータセットの中で，元画像と形が似た建物がアパートメントの割合が多かったことが原因であると考える．一方，図 5.14 の結果から，元画像と生成画像が一軒家で一致している．これは，元画像の建物の形が学習させたヨーロッパのデータセットの中で，元画像と形が似た建物が一軒家の割合が多かったことが原因であると考えら

れる，また，窓の生成には成功した.

第 6 章　おわりに

6.1　まとめ

本研究では，Google Street View から取得したヨーロッパの街並みと日本の空き家の画像を使って実験を行い，日本の空き家の外観デザイン変更の一例としてヨーロッパのレンガ風の建物に改修できることを確認した.

6.2　今後の課題と展望

今後の課題としては，　ヨーロッパのレンガの建物という広いカテゴリのデータセットのみの学習だけでは，建物の細かい要素のデザイン生成が困難であると考えられる．そのため，さらに，窓の形や屋根の形などのディテールも統一させて学習する，または窓や屋根，ドアなどの建物の要素ごとのマスク画像を作成し，学習することでより精緻な画像を生成ができると考えている．また，Google Street View を用いた画像取集においては，現段階では，目視で学習に使う画像を区別している．今後は，Google Maps API を使用すれば，自動で建物の正面の画像や全体が写った画像のみを収集できるようにし，完全自動化を目指す.

謝辞

本研究を進めるにあたって，ご協力いただいた建築学科の Rami 教授，保澤伸光さん，Cliff さんをはじめ，同研究室の学生の皆様ありがとうございました．建築学科の方々には，とても有益な多くの助言をいただきました．また，指導教員の山口准教授には，多くの専門的なアドバイスに加え，研究に対する姿勢やノウハウから教えていただき，とても感謝しております．ありがとうございました．

参考文献

[1] 総務省(2024.4.30).「令和５年住宅・土地統計調査　住宅数概数集計（速報集計）結果」,
(参照 2025-01-02)
https://www.stat.go.jp/data/jyutaku/2023/pdf/g_kekka.pdf

[2] やまとごころ.jp(2017.03.23),「古民家の宿に高まる外国人の関心、 人気の古民家宿のストーリーとは？(後編)」,（参照 2025-01-02)
https://yamatogokoro.jp/report/3373/

[3] ハヤトリフォーム(2024.07.08),「外壁塗装の色選び！カラーシミュレーションを使った成功の秘訣！ -春日市の外壁塗装ならハヤトリフォームへ！」,（参照 2025-01-02)
https://nurikae.jp/%E5%A4%96%E5%A3%81%E5%A1%97%E8%A3%85%E3%81%AE%E8%89%B2%E9%81%B8%E3%81%B3%EF%BC%81%E3%82%AB%E3%83%A9%E3%83%BC%E3%82%B7%E3%83%9F%E3%83%A5%E3%83%AC%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%82%92%E4%BD%BF/

[4] 金丸塗装,「外壁・屋根塗装のカラーシミュレーションと色選びのポイント」,
(参照 2025.01-02)
https://www.kanemaru-tosou.jp/base/color-simulation/

[5] 株式会社インディゴ,「3DCG はなぜ難しい？初心者から卒業する方法と学び方を紹介」
（参照 2025.01.02)
https://www.indigo-studio.jp/2023/01/27/3dcg-difficult/

[6] 山田悟史，大野耕太郎(2020),「Deep Learning を用いたデザイン AI の作成と検証――街並みと建築物外観の画像生成を対象に」,（参照 2025.01.05),日本建築学会　計画系論文集，第 85 巻，第 770 号

[7] 金子卓弘(2018),「日本電信電話株式会社，Generative Adversarial Networks の基礎と応用」,（参照 2025.01.05),日本音響学会誌 74 巻 4 号 (2018),pp.208-218.

[8] 福岡知隆，南貴大，浦田渡，藤生慎，高山純一(2019),「深層学習による橋梁点検のための Pix2Pix による疑似訓練データ作成」,（参照 2025.01.05),土木学会論文集 F4(建設マネジメント),Vol.75,No.2,1_27-1_35,2019.

[9] Shutterstock,「空き家 日本のロイヤリティフリー画像」,(参照 2024.12.28)
https://www.shutterstock.com/ja/search/%E7%A9%BA%E3%81%8D%E5%AE%B6-%E6%97%A5%E6%9C%AC

[10] 不動産売却コラム 八城地建(2023.06.30),「空き家の固定資産税が 6 倍になるのはいつから？法改正で対象が増える！」,(参照 2024.12.28)
https://www.8shiro.jp/sell-column/cost/870/