

# 機械学習向けCGデータの 量産手法の検討

シリコンスタジオ株式会社

技術本部

佐藤 晶威

masatake.sato@siliconstudio.co.jp

---

## 本講演の内容

- 機械学習におけるCG技術の需要
- CG制作のアプローチと課題
- 画像品質合わせ込みの自動化検討
- レゴブロックを用いた実写とCGの品質比較検証

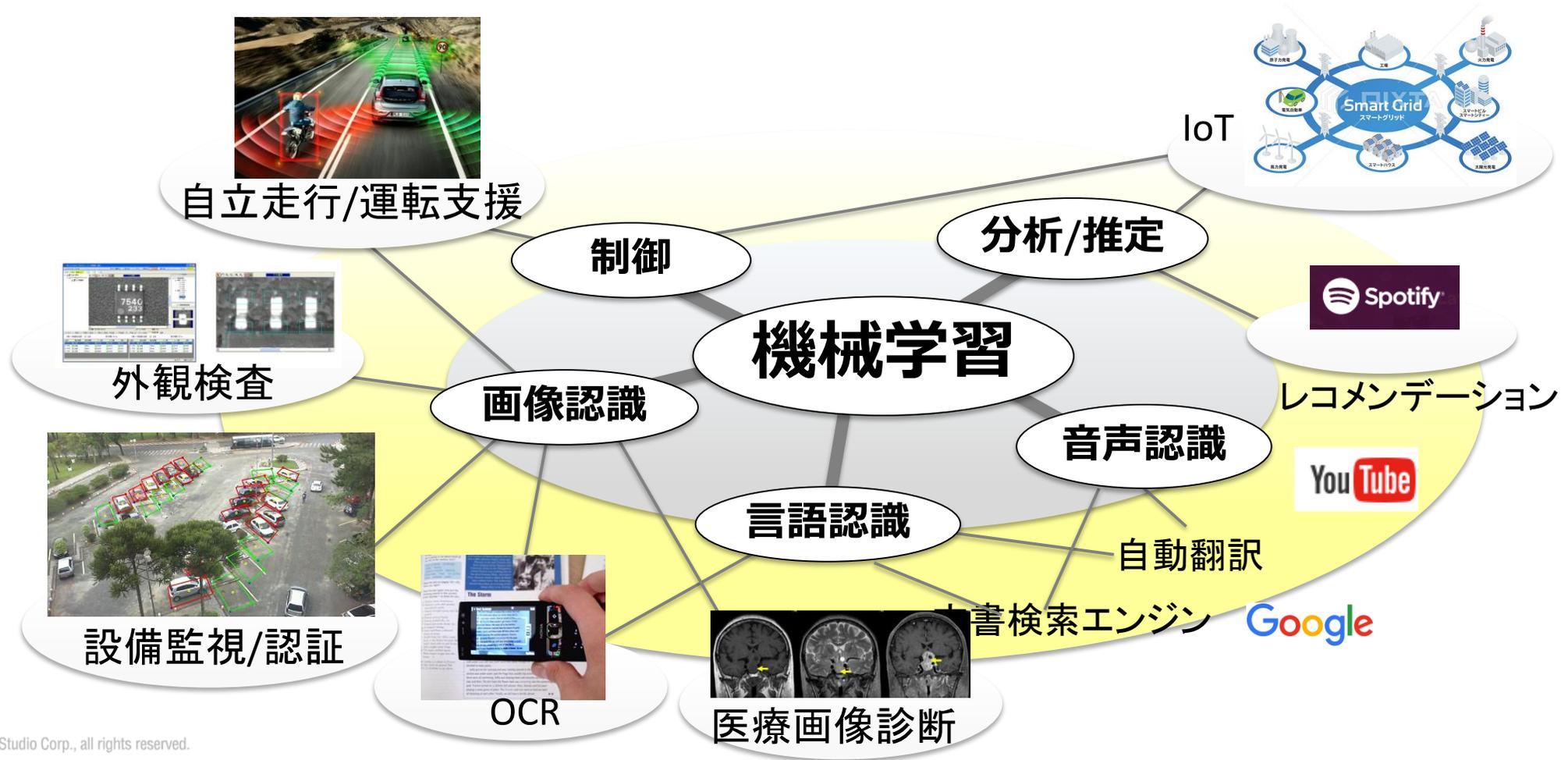
# もくじ

- 背景と目的
- CG制作のアプローチ
- 課題
- 解決方法
- 実現手段
- 実験と結果
- まとめ

## シリコンスタジオのビジネス戦略

- 産業分野を越えて活用できるソリューション/技術の提供
  - ゲーム産業、アニメ産業
  - 製造業、広告業
- グラフィックス技術を転用した新規サービスの開拓
  - 広告販促向けCG
  - 組み込みグラフィックス
  - 機械学習

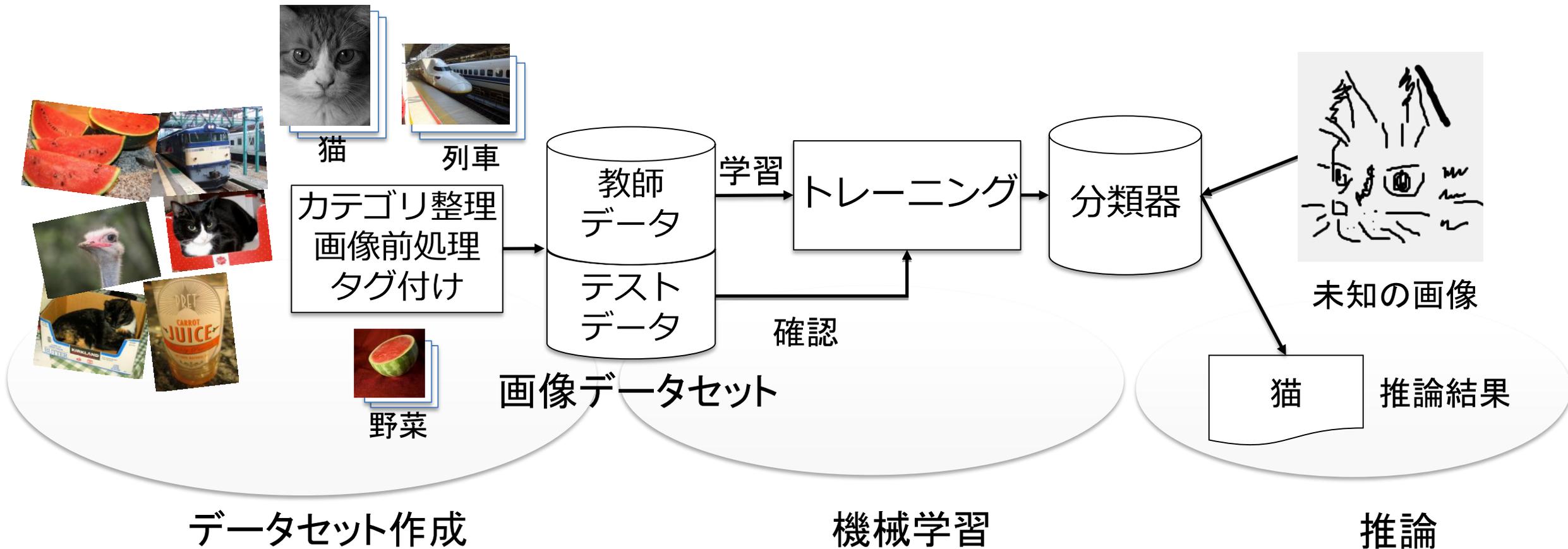
# 機械学習を活用したサービス



## 注目分野と技術開発の方向性

- 画像認識を中心とした機械学習技術
  - 外観検査
  - 設備監視/認証
  - 自立走行/運転支援
  - OCR
- 機械学習におけるコンピュータグラフィックス技術活用
  - 学習用CG画像データの作成
  - 評価用シミュレーション環境の構築

# 画像認識における機械学習の基本的な流れ



## 画像データセットの特徴

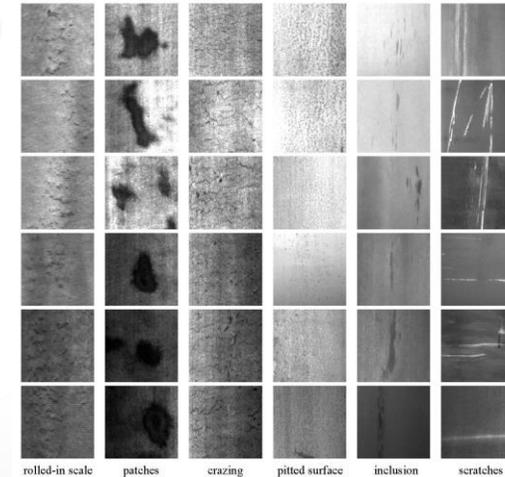
- 一定規模の画像データが必要
  - 数千枚～数万枚
  - タグ情報付き
- 事象の網羅性も重要
  - 同じ場所で天候が異なる
  - 発生頻度が低い状況



PubFig83 + LFW Dataset



CompCars dataset



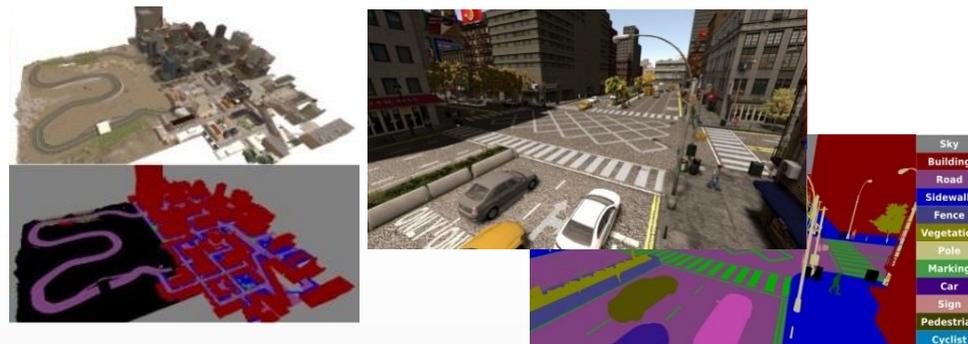
NEU surface defect database

## 機械学習におけるCG活用事例

- 自動運転向け（物体認識用途）の事例
  - Virtual KITTI、SYNTHIA
  - VERTECHs
- 画像データセットをCGで作成
  - 実測よりも低コストに量産
  - 多様なバリエーション
  - 実写画像に準じた画質



Virtual KITTI dataset



SYNTHIA Dataset

## 機械学習向けCG制作のための仕組み作り

- ~2016年
  - リアルタイムグラフィックスを活用した画像データ生成
    - 持ち合わせのツールで最大限の品質で制作
- 2017年~
  - CG画像量産化のための制作手法検討
  - 学習に効果的な品質への合わせ込み手法検討

目的：少量の実写画像サンプルに基づく学習用CG画像データ量産  
(数千枚~数万枚のバリエーション画像)

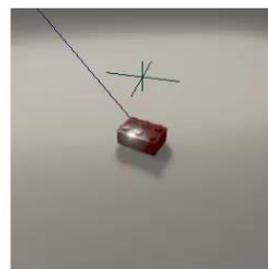
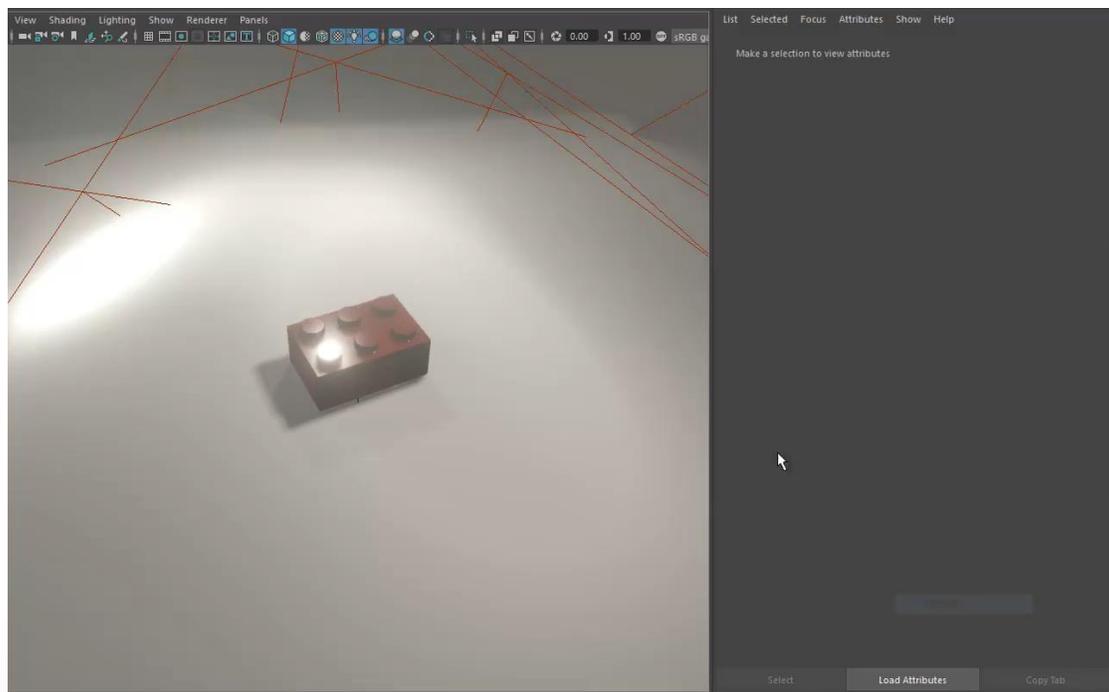
# もくじ

- 背景と目的
- **CG制作のアプローチ**
- 課題
- 解決方法
- 実現手段
- 実験と結果
- まとめ

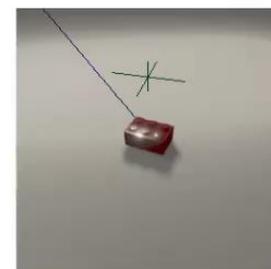
## CG制作のアプローチ

- 画像データを低コストに量産する
  - リアルタイムグラフィックスの活用
- バリエーションを表現する
  - プロシージャル生成技術の活用
- 実写画像に準じた品質にCG画像を合わせ込む
  - パラメータチューニングの実施

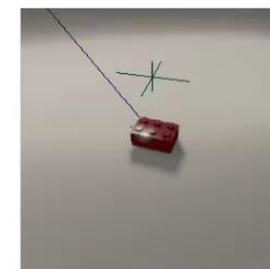
# リアルタイムグラフィックスの活用



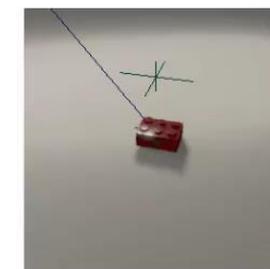
img\_000000000.png



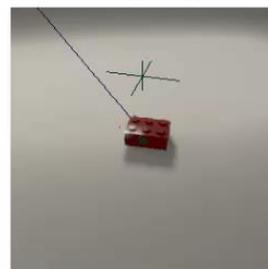
img\_000000001.png



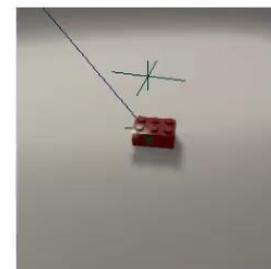
img\_000000002.png



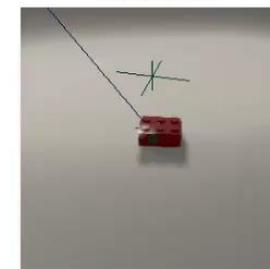
img\_000000003.png



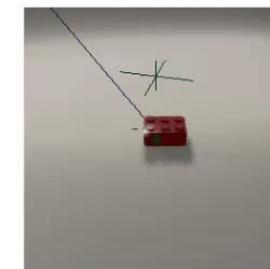
img\_000000004.png



img\_000000005.png



img\_000000006.png



img\_000000007.png

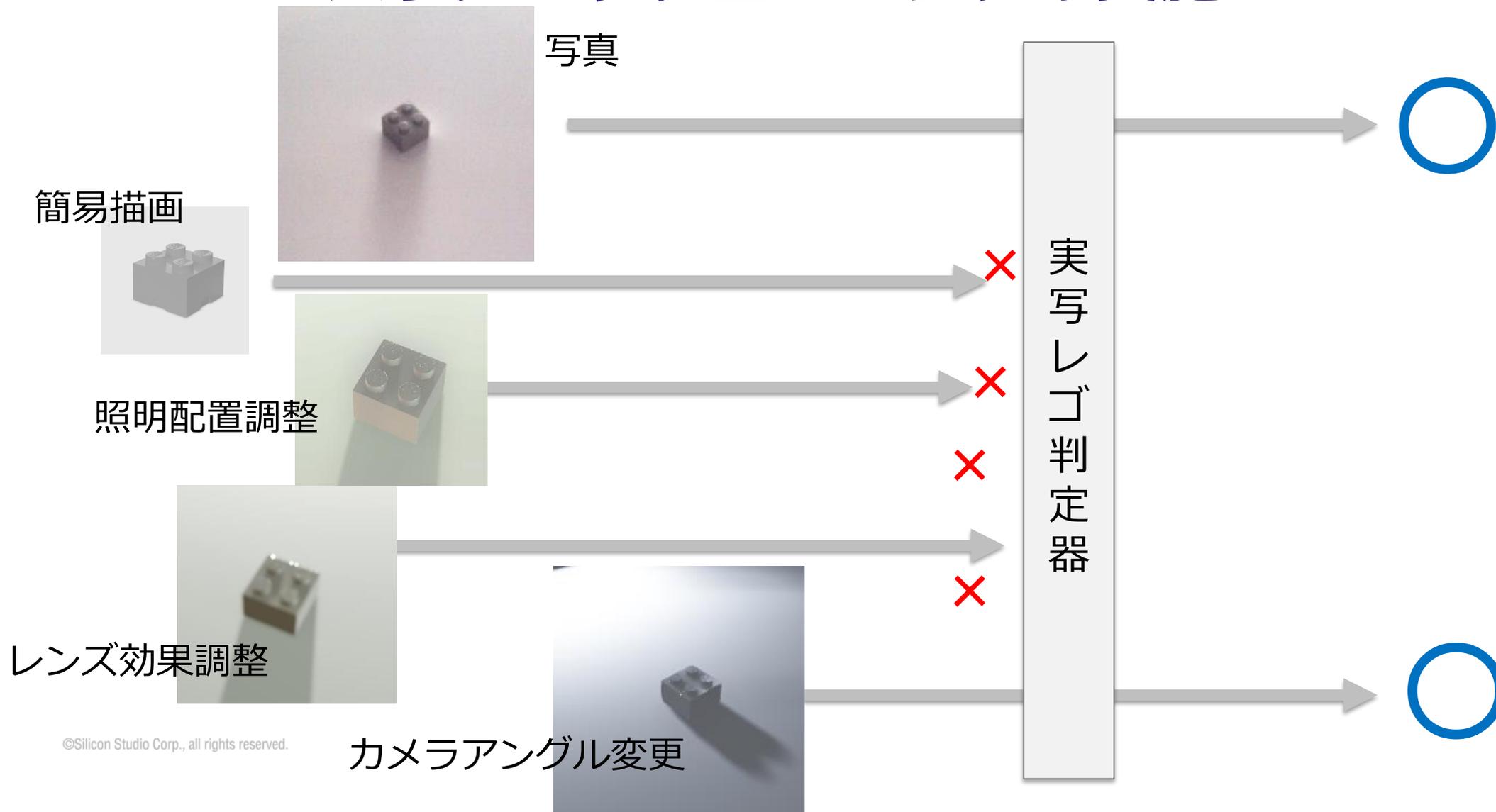
# プロシージャル生成技術の活用



# プロシージャル生成技術の活用



# パラメータチューニングの実施

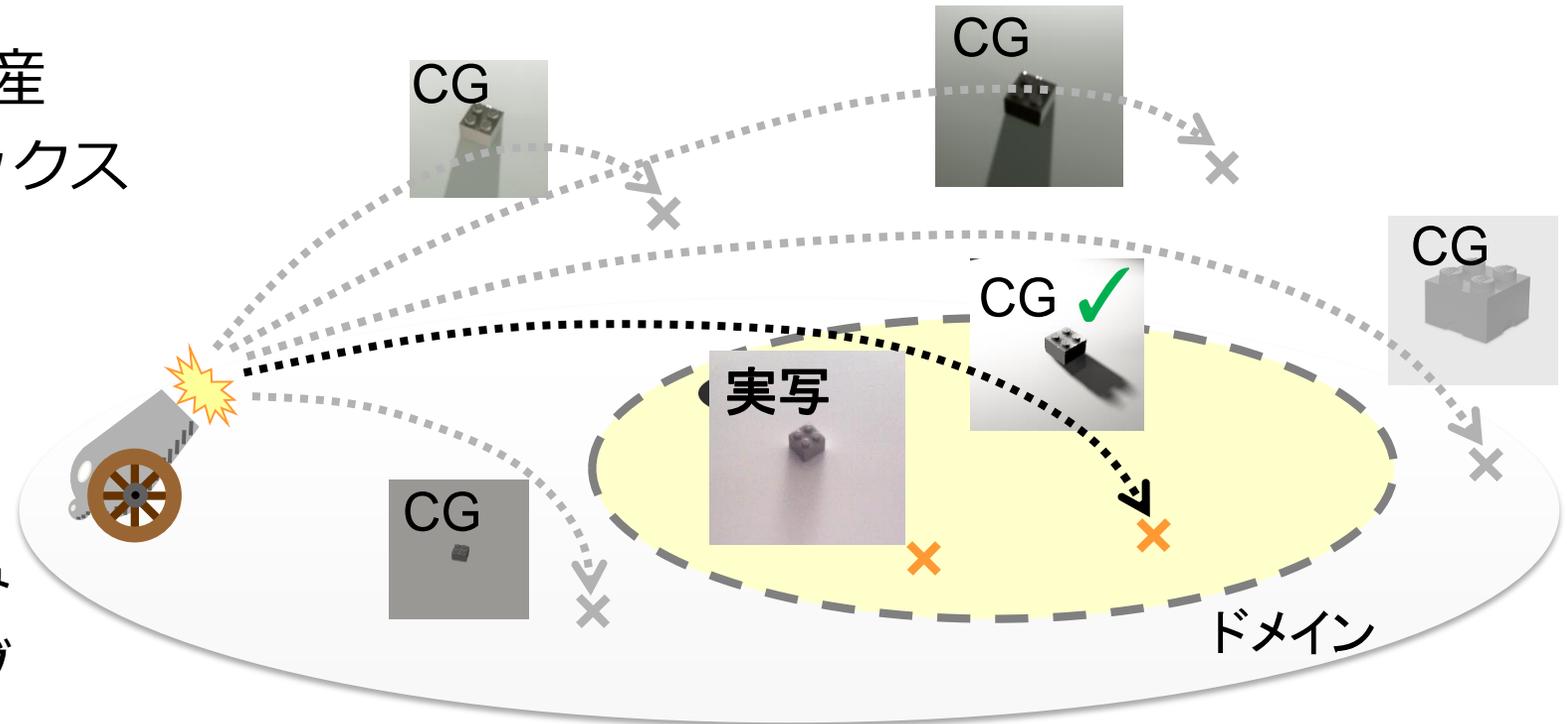


# もくじ

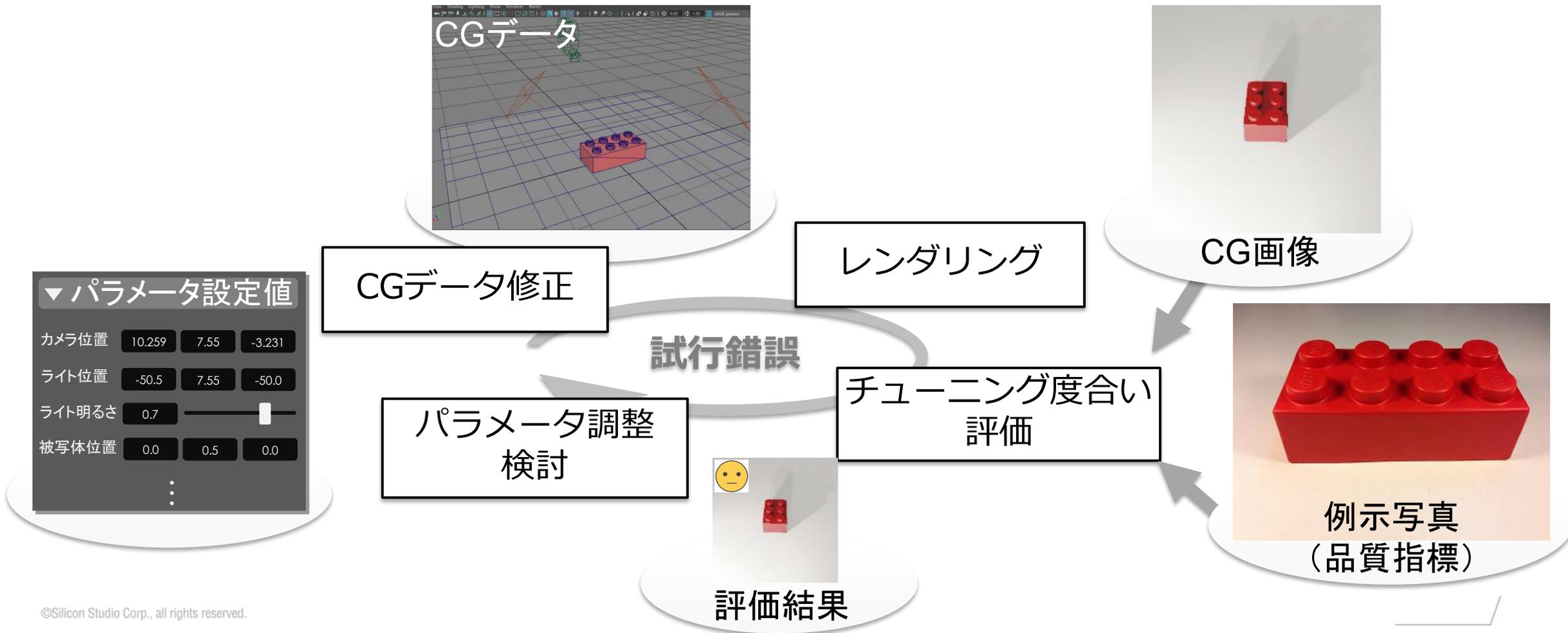
- 背景と目的
- CG制作のアプローチ
- **課題**
- 解決方法
- 実現手段
- 実験と結果
- まとめ

## 機械学習向けCG制作の状況

- ✓ 画像データの低コスト量産
  - リアルタイムグラフィックス
- ✓ バリエーションの表現
  - プロシージャル生成
- ✗ CG画像品質の合わせ込み
  - パラメータチューニング
    - 時間を要する



# 現状のパラメータチューニングの流れ



## パラメータチューニングの課題

- 主観によるチューニング度合いの評価
  - 制作初期の要件決めをする上では重要
  - 合わせ込みはより明確な指標が必要
- 試行錯誤の複雑化
  - パラメータ数の増加に伴う試行回数増加
  - 実験の網羅性

**試行錯誤を網羅的に素早くこなす手段が必要**

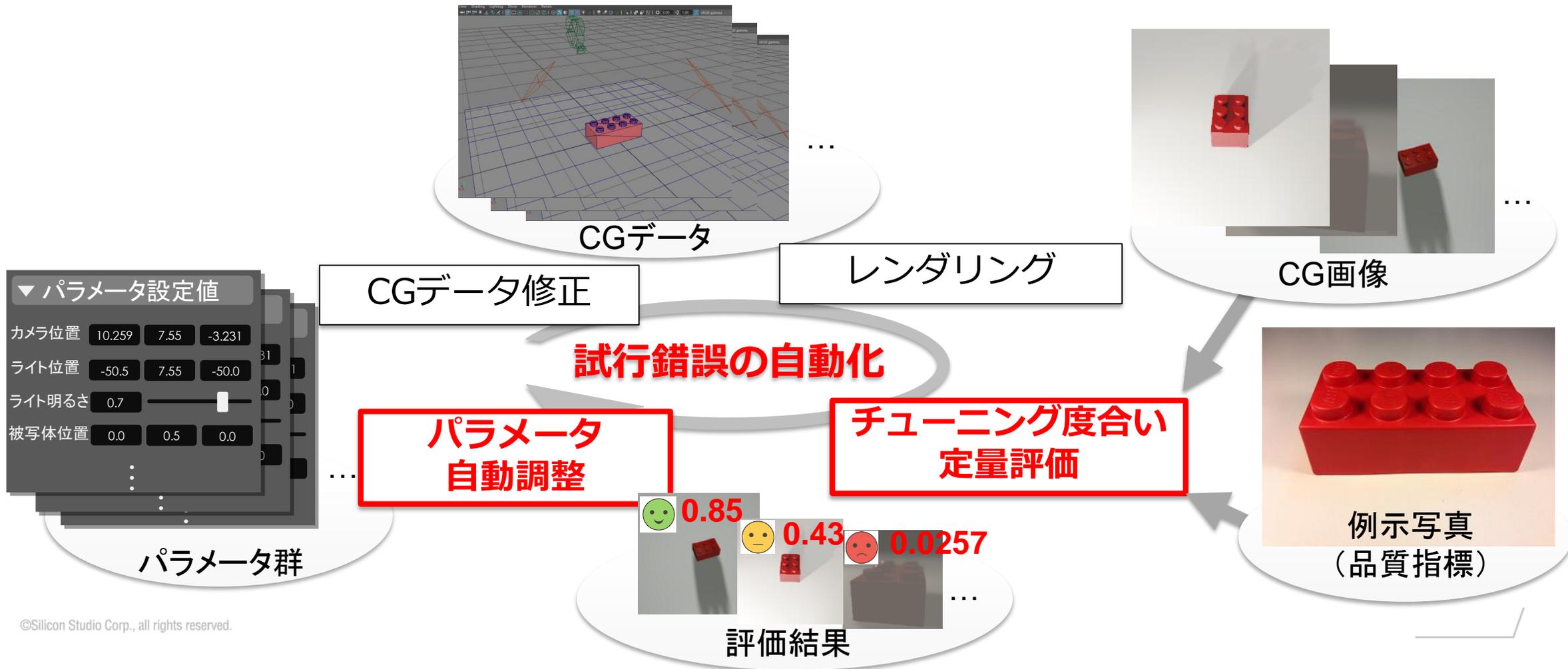
# もくじ

- 背景と目的
- CG制作のアプローチ
- 課題
- **解決方法**
- 実現手段
- 実験と結果
- まとめ

# パラメータ自動チューニング技術の開発

- チューニング度合いの定量評価
  - 目標品質を明確化する
  - 目標品質に対するズレを客観的に把握する
- 試行錯誤の自動化
  - パラメータ探索の網羅性を維持して取りこぼしを減らす
  - 高感度パラメータの高精度な合わせ込み

# パラメータ自動チューニングの流れ

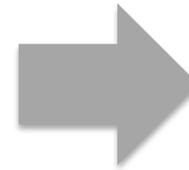


# もくじ

- 背景と目的
- CG制作のアプローチ
- 課題
- 解決方法
- **実現手段**
- 実験と結果
- まとめ

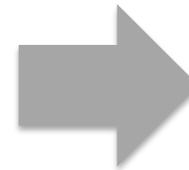
## パラメータ自動チューニングの要素技術

- チューニング度合いの定量評価
  - 例示写真とCG画像の類似度で評価



画像特徴量を用いた  
類似度計算

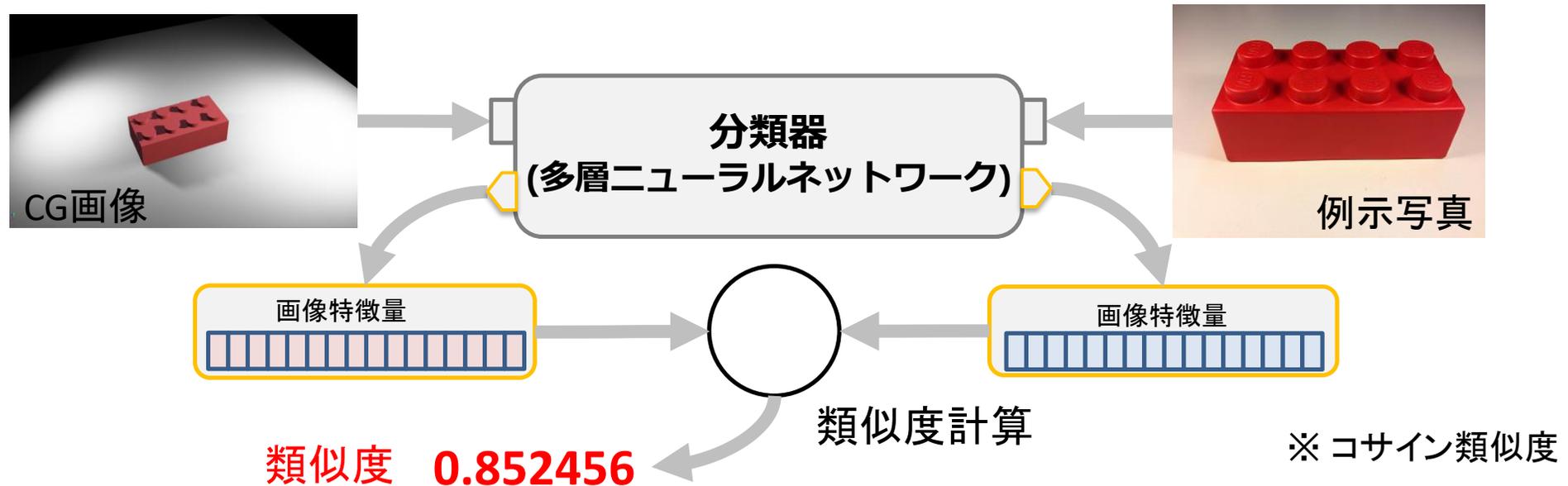
- 試行錯誤の自動化
  - 評価結果に基づくパラメータ自動調整



最適化問題を用いた  
パラメータの自動探索

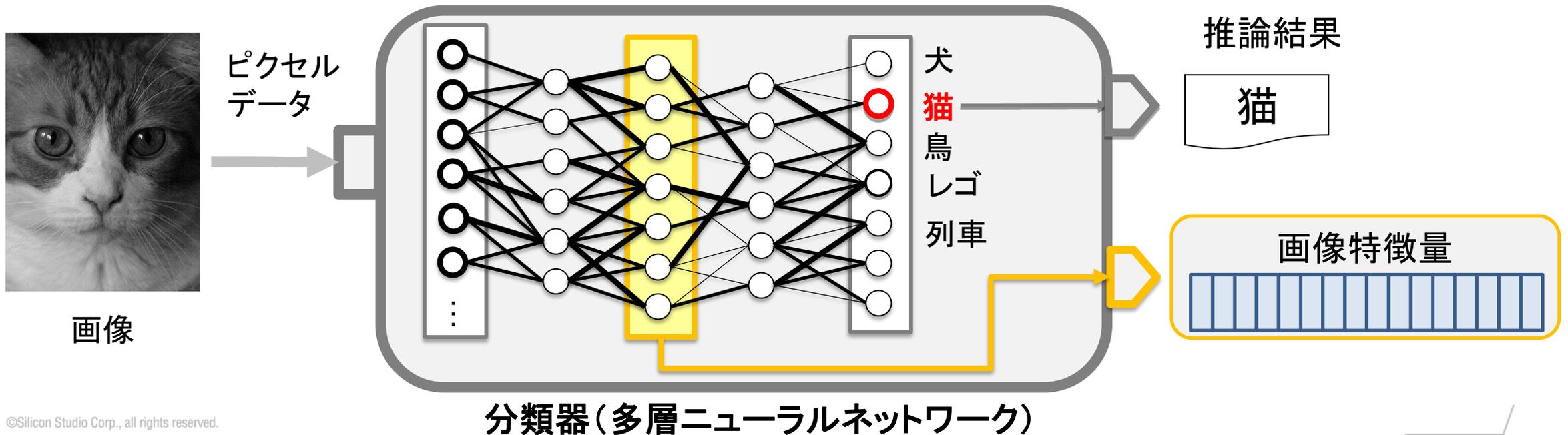
## チューニング度合いの定量評価

- 分類器から画像特徴量を抽出
- 特徴量間の類似度※を「チューニング度合い」として使用

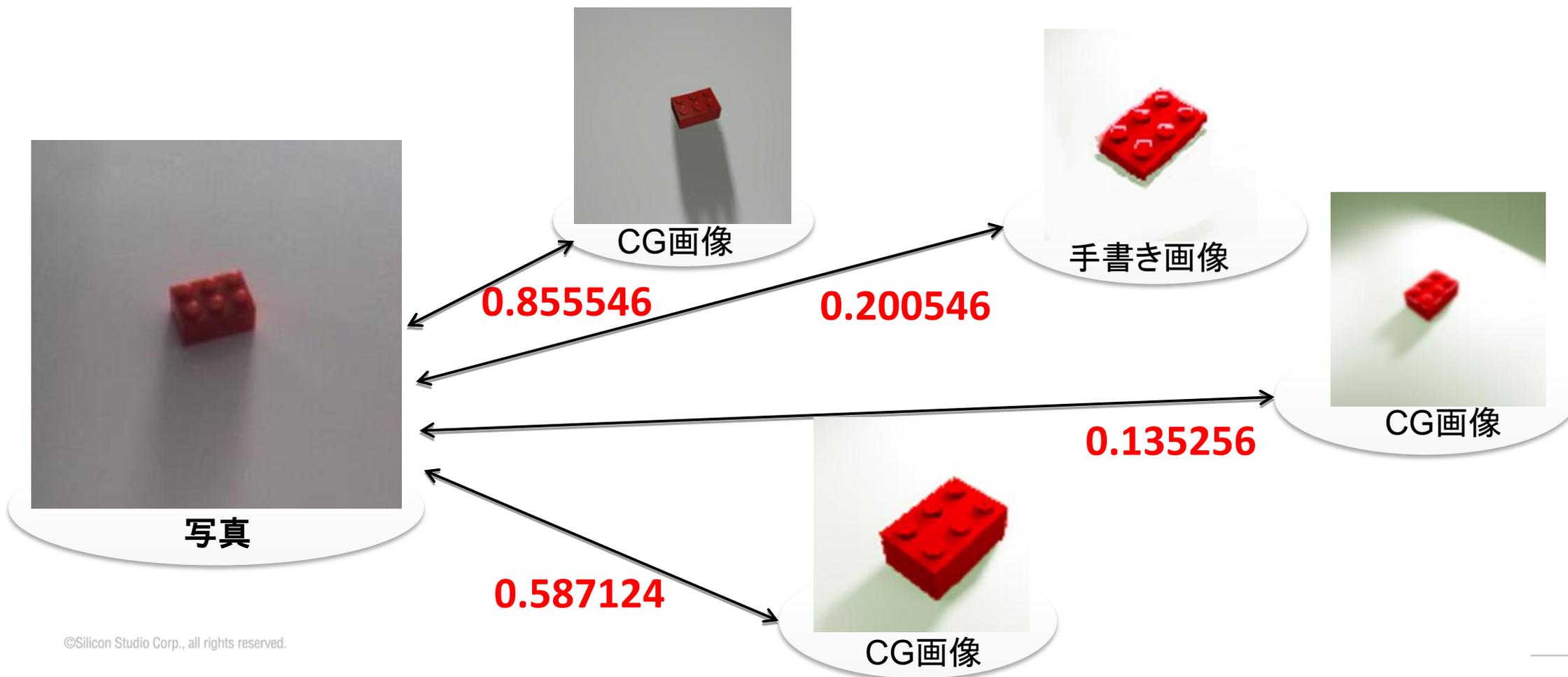


## チューニング度合いの定量評価

- Deep Features [B. Zhou, 2014]
  - 分類器の中間層出力を画像特徴量として使用



# チューニング度合い定量化の例

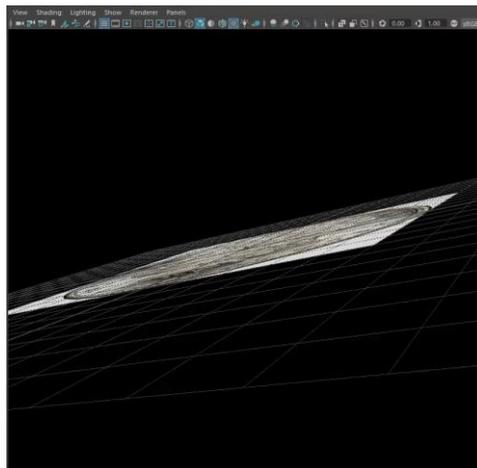


## 試行錯誤の自動化手法

- 最適化問題を用いたパラメータ自動探索
  - 進化的アルゴリズム※を使用



例示画像

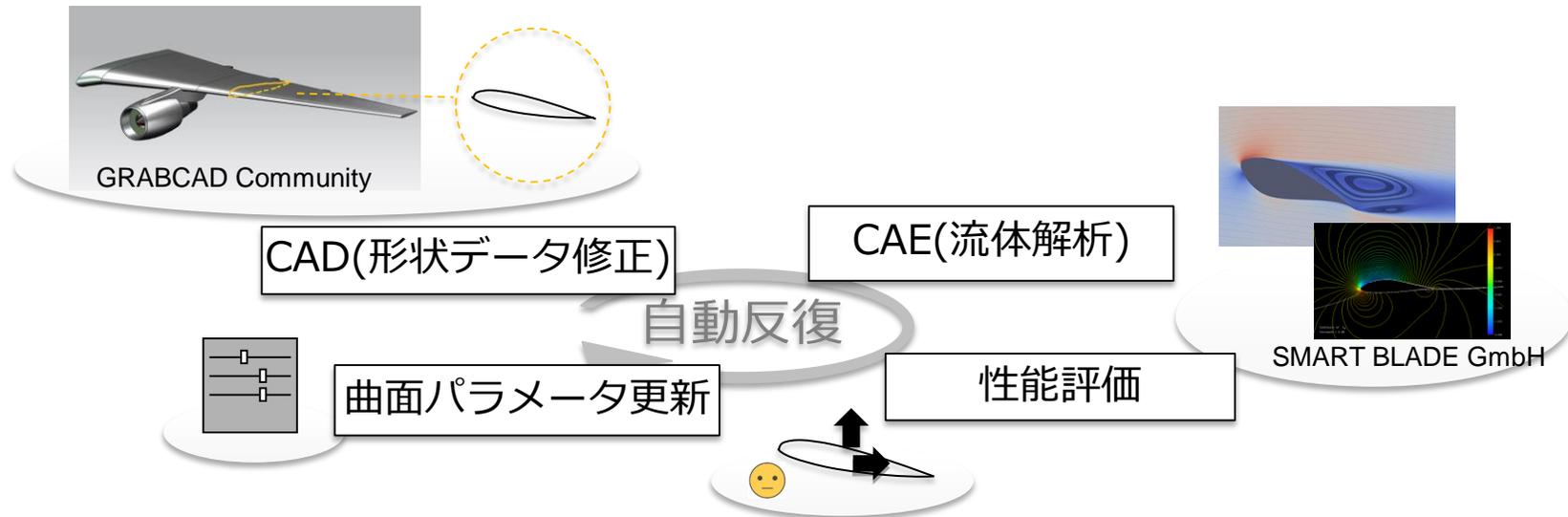


CGデータ

カメラ撮影位置の自動探索例

## 試行錯誤の自動化手法

- 最適化問題を用いたパラメータ探索手法を採用
  - CAD/CAE等グラフィックス関連の技術/手法を調査 (90's~)



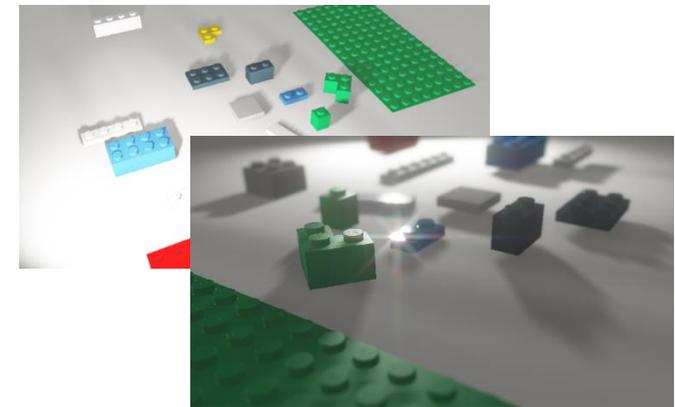
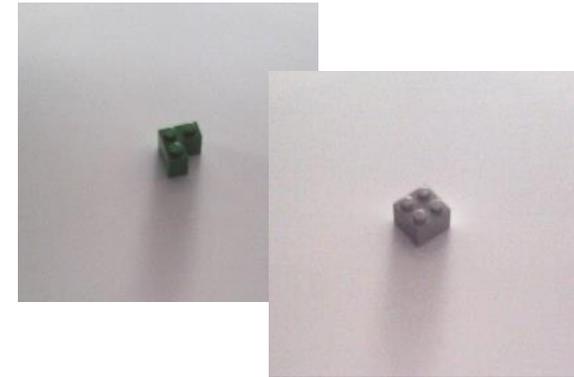
翼断面の形状最適化(空力特性の改善)の例

# もくじ

- 背景と目的
- CG制作のアプローチ
- 課題
- 解決方法
- 実現手段
- **実験と結果**
- まとめ

## 実験概要

- 目的
  - パラメータ自動チューニングの効果検証
- 実施内容
  - 例示写真の品質にCG画像を近づける
- 実験対象
  - 外観検査を想定したシンプルな対象からスタート
  - レゴブロック
    - 製造ばらつきの少なさ、入手性高、撮影簡単
    - CGデータ制作コストも低



# 実験

- 事前準備
  - レゴCGデータの作成
  - レゴ分類器の学習（チューニング度合い定量評価用）
- 実験結果

# チューニング対象パラメータ

- レンダリング品質に与える影響が顕著で、編集頻度の高いものをピックアップ(全53変数)

- レゴブロックの位置

- カメラの位置

- ライトの位置、明るさ、大きさ

空間配置パラメータ

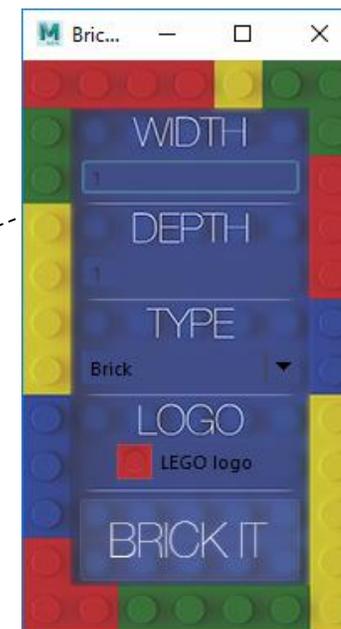
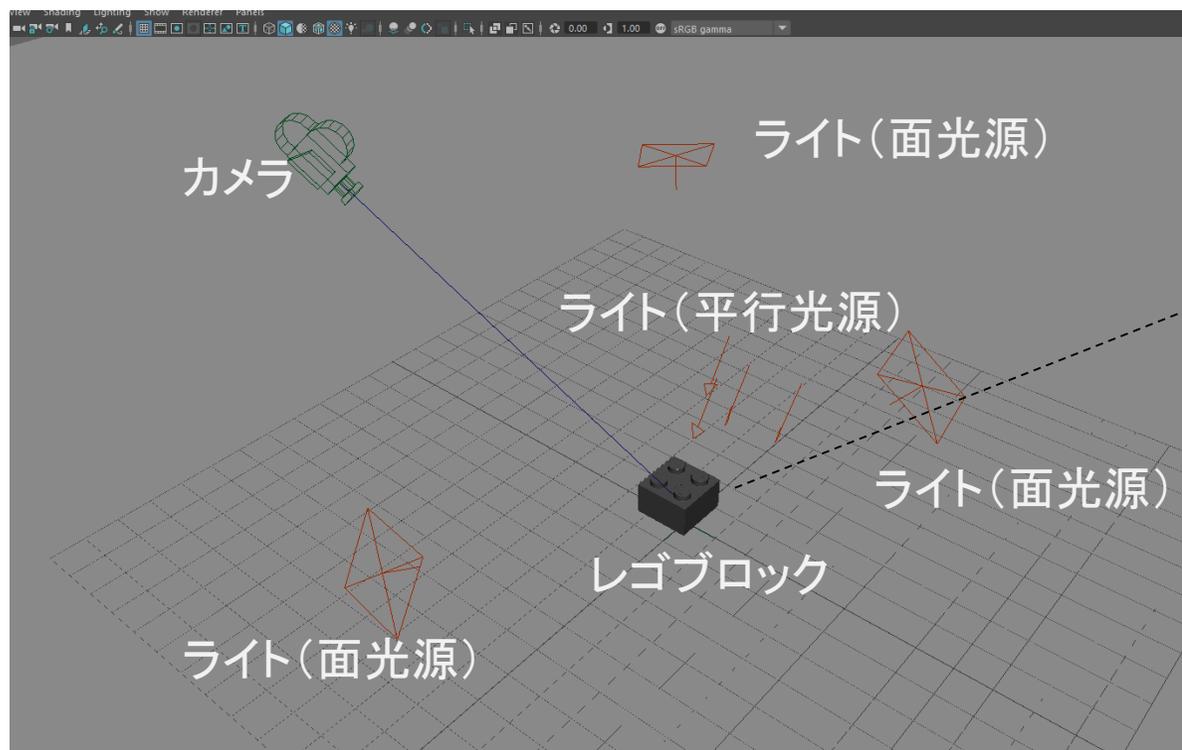
- 被写界深度（焦点距離、F値等）

- 露光、色相、彩度、明度、ホワイトバランス、ガンマ

撮影カメラ特性パラメータ

## 空間配置パラメータ

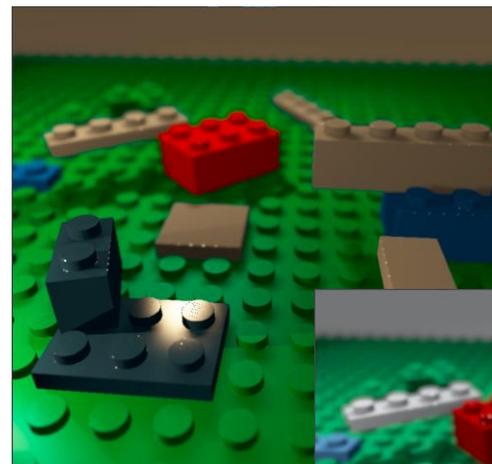
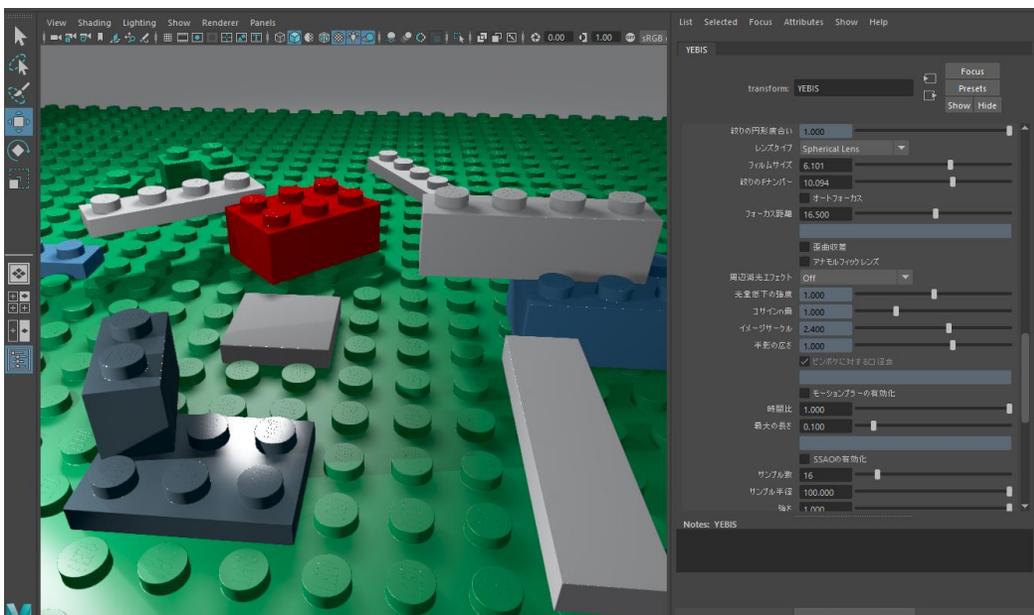
- Maya 2017で仮想撮影環境を構築



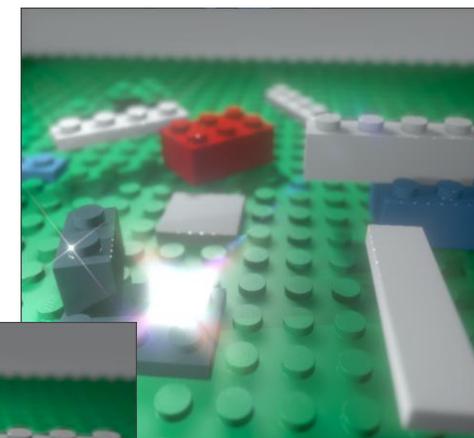
Brick Generator &  
Build A Brick

# カメラ特性パラメータ

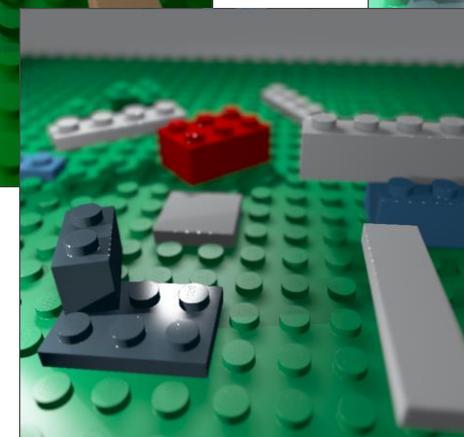
- YEBIS for Mayaを使って近似



色相変更



グレア効果



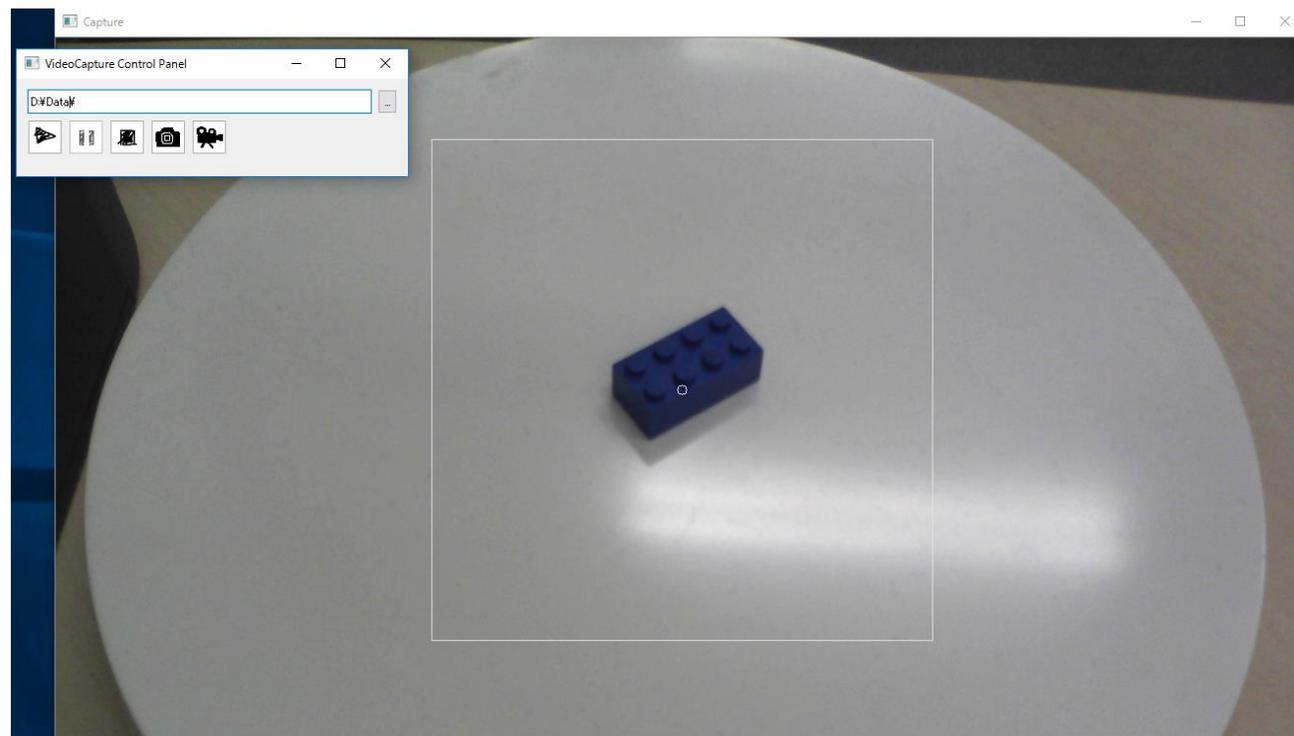
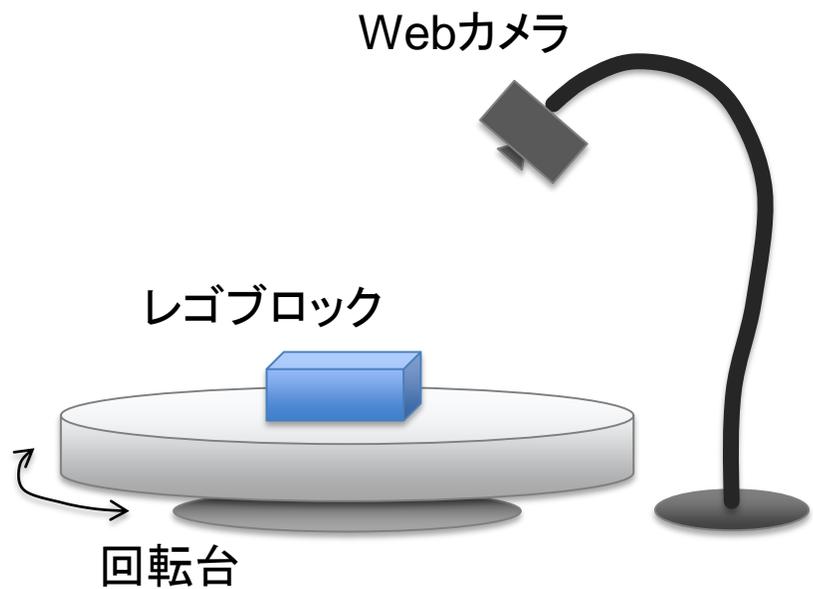
レンズボケ効果

# 実験

- 事前準備
  - レゴCGデータの準備
  - レゴ分類器の学習（チューニング度合い定量評価用）
- 実験結果

## まずレゴ写真を撮影

- レゴブロックの撮影環境

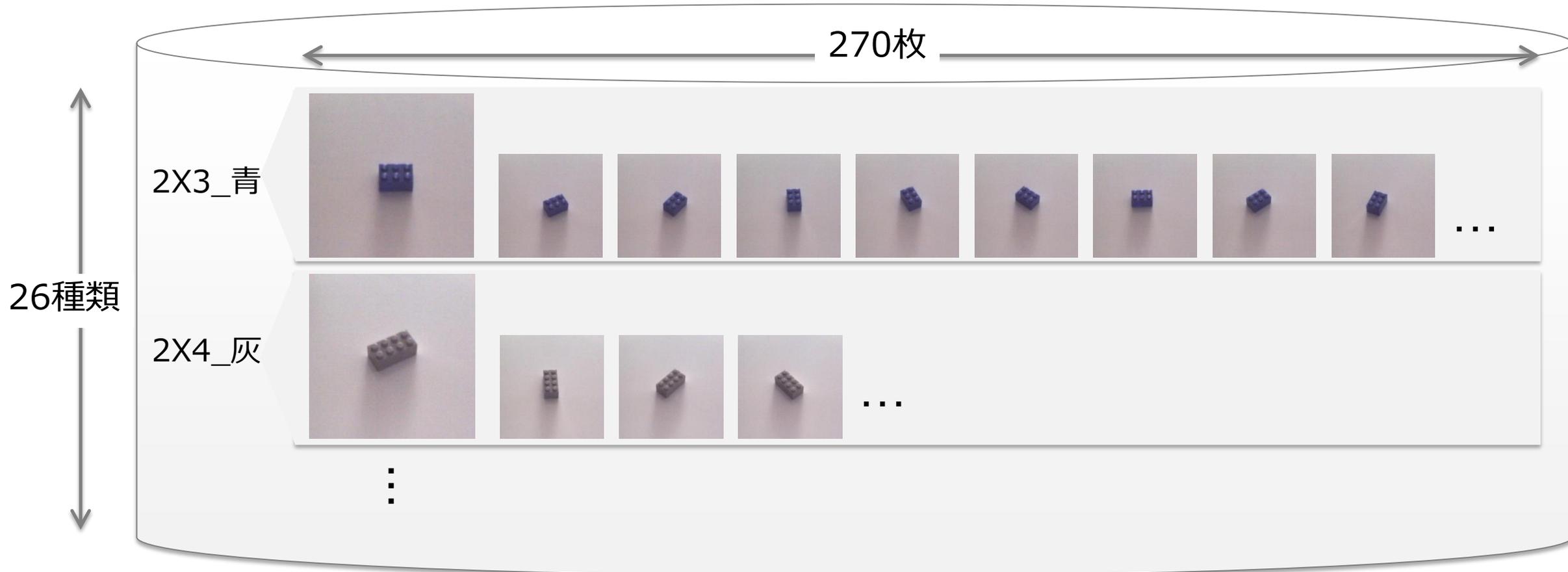


# レゴ写真データセットのカテゴリ

- 26種類のレゴブロックを選定

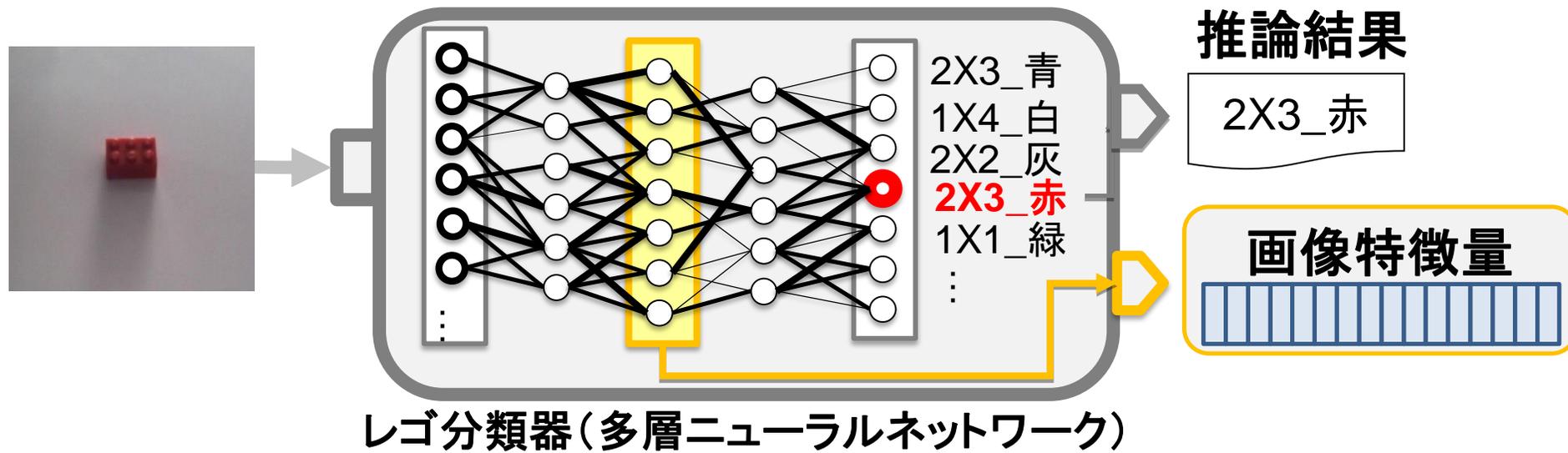


# レゴ写真データセットの構成



# レゴ写真データセットを使って分類器を学習

- レゴブロックの種類を判別する多層ニューラルネットワーク
  - Deep Learningで学習 (ImageNet学習済みモデルからファインチューニング)



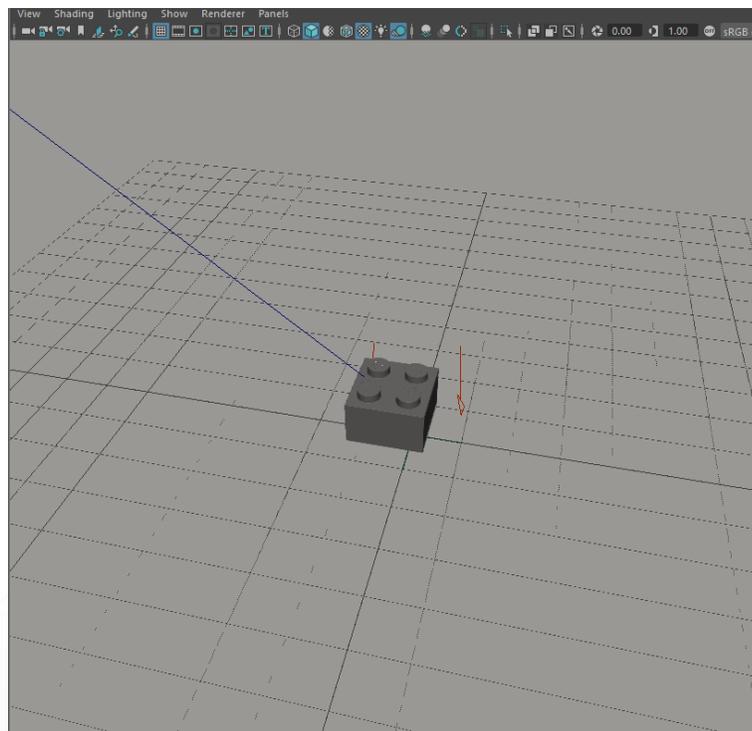
# 実験

- 事前準備
  - レゴCGデータの準備
  - レゴ分類器の学習（チューニング度合い定量評価用）
- 実験結果

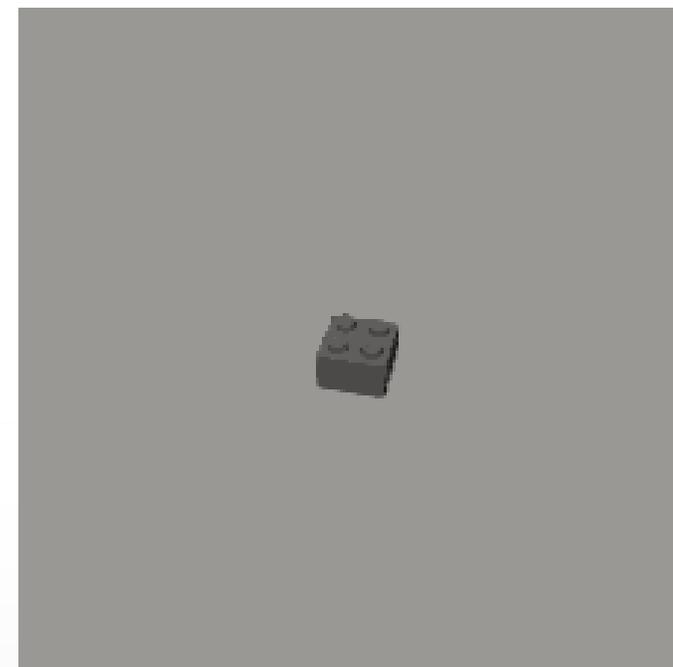
# パラメータ自動チューニング実験 1



例示画像

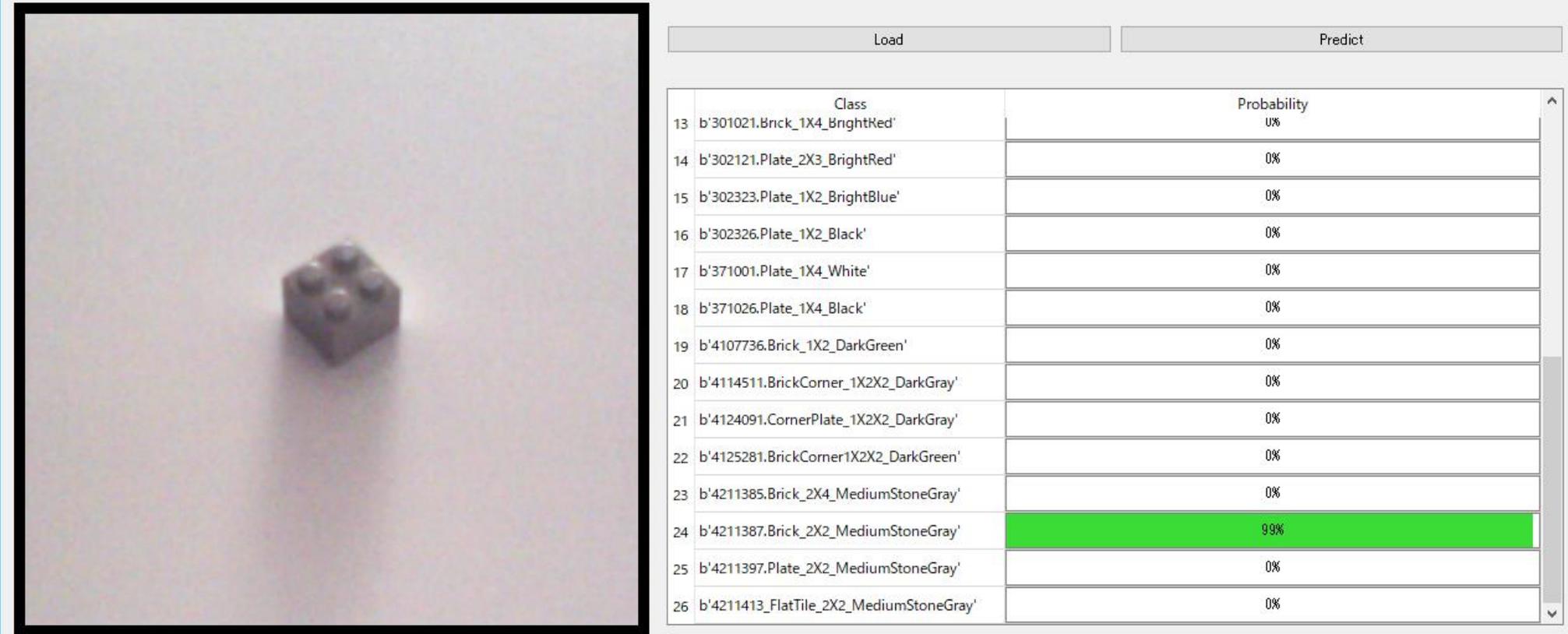


レゴCGデータ(初期状態)



CG画像

# 例示画像の推論結果



The screenshot displays the 'Classify' window in Silicon Studio. On the left, a photograph of a dark gray brick is shown. On the right, a table lists classification results for various brick classes. The class 'b'4211387.Brick\_2X2\_MediumStoneGray' is highlighted in green, indicating a 99% probability.

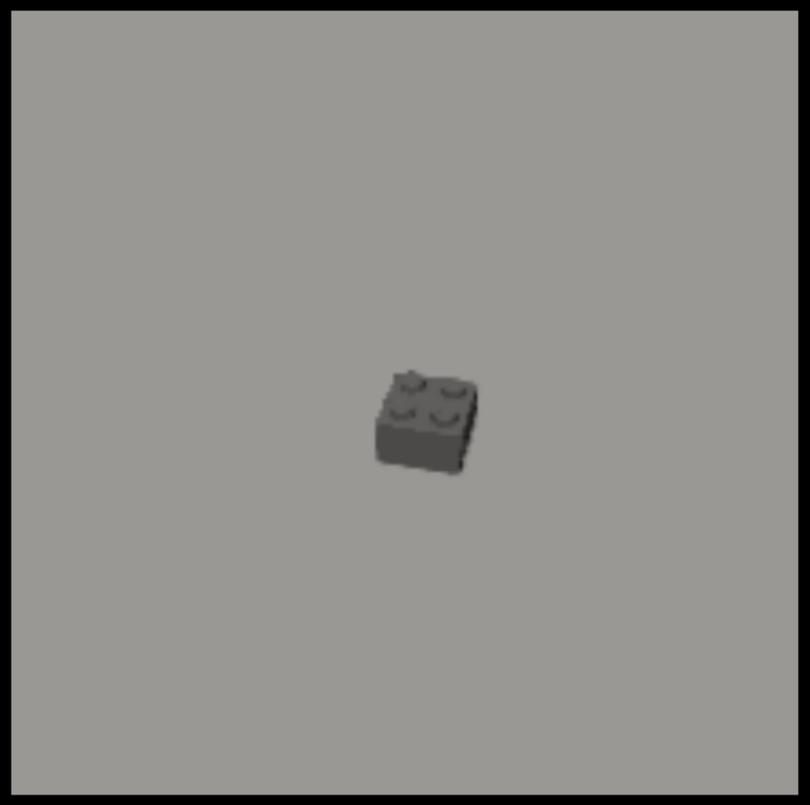
	Class	Probability
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%
15	b'302323.Plate_1X2_BrightBlue'	0%
16	b'302326.Plate_1X2_Black'	0%
17	b'371001.Plate_1X4_White'	0%
18	b'371026.Plate_1X4_Black'	0%
19	b'4107736.Brick_1X2_DarkGreen'	0%
20	b'4114511.BrickCorner_1X2X2_DarkGray'	0%
21	b'4124091.CornerPlate_1X2X2_DarkGray'	0%
22	b'4125281.BrickCorner1X2X2_DarkGreen'	0%
23	b'4211385.Brick_2X4_MediumStoneGray'	0%
24	b'4211387.Brick_2X2_MediumStoneGray'	99%
25	b'4211397.Plate_2X2_MediumStoneGray'	0%
26	b'4211413.FlatTile_2X2_MediumStoneGray'	0%

# パラメータ自動チューニング前のCG画像推論結果

Classify

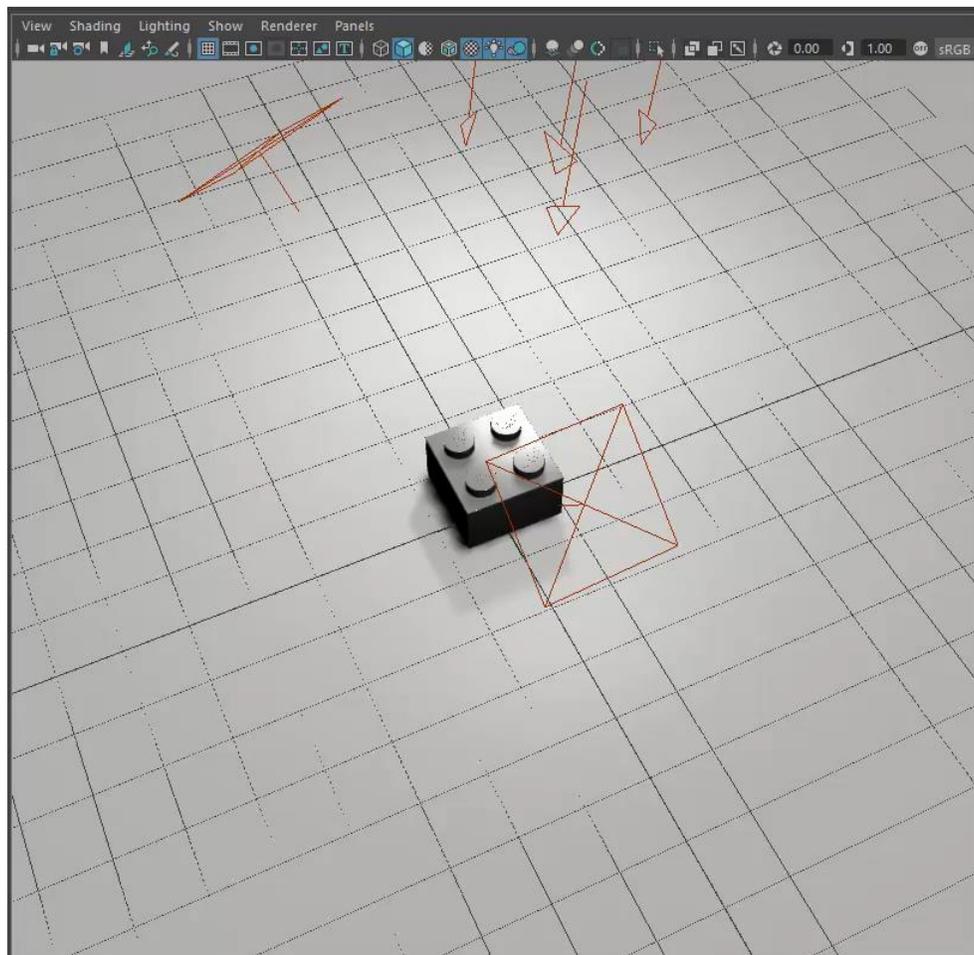
Load Predict

	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	0%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	3%
8	b'300423.Brick_1X2_BrightBlue'	2%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	85%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%



©Silicon Studio Corp., all rights reserved.

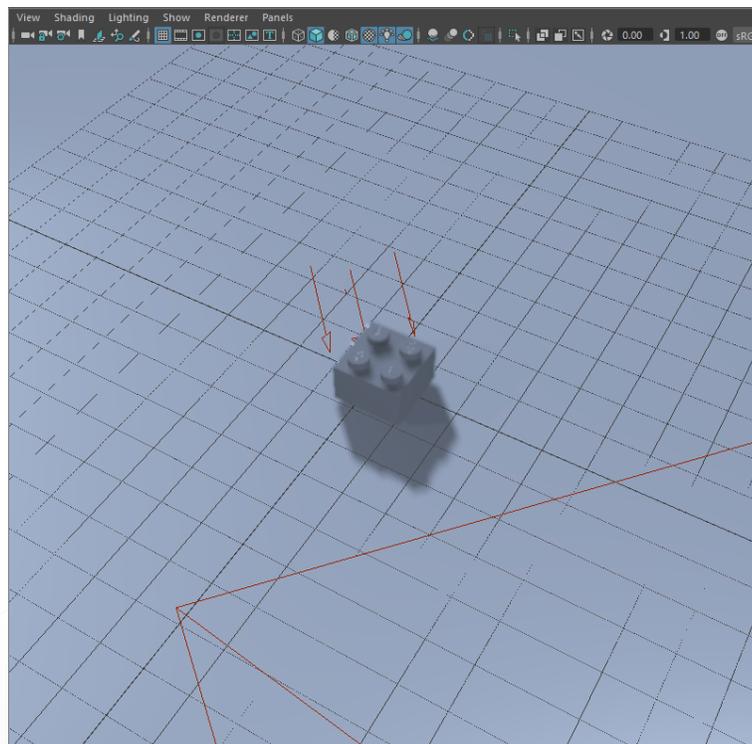
# パラメータ自動チューニングの様子



# パラメータ自動チューニング結果



例示画像

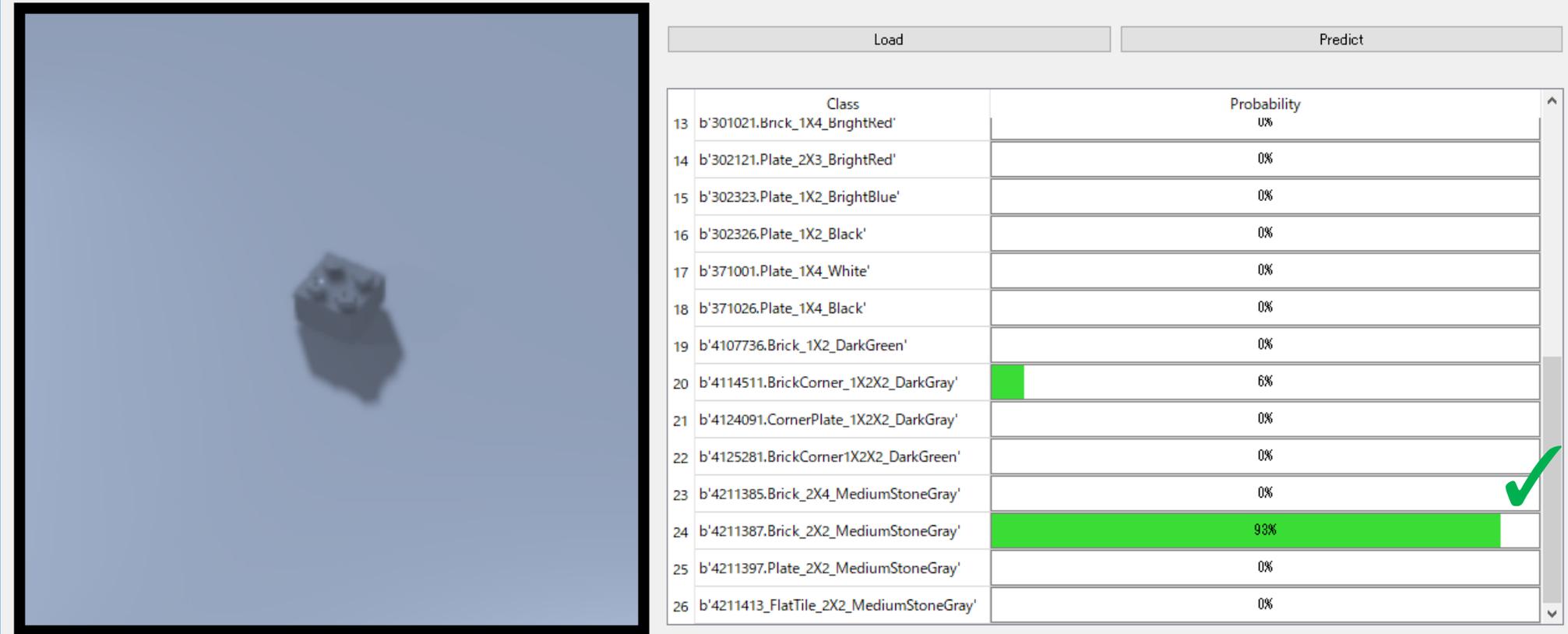


レゴCGデータ



CG画像

# パラメータ自動チューニング後のCG画像推論結果



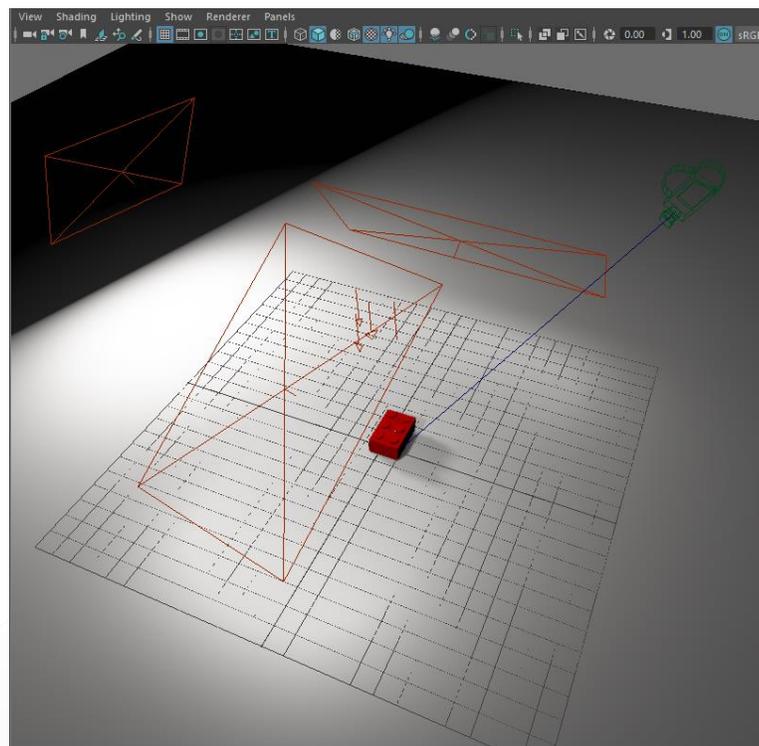
The screenshot shows the 'Classify' window in Silicon Studio. On the left, a 3D model of a brick is displayed on a light blue background. On the right, there is a table with two columns: 'Class' and 'Probability'. The table lists 13 different brick classes. The class 'b'4211387.Brick\_2X2\_MediumStoneGray' has a probability of 93% and is highlighted in green. A green checkmark is placed to the right of this row. The 'Predict' button is visible above the table.

	Class	Probability
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%
15	b'302323.Plate_1X2_BrightBlue'	0%
16	b'302326.Plate_1X2_Black'	0%
17	b'371001.Plate_1X4_White'	0%
18	b'371026.Plate_1X4_Black'	0%
19	b'4107736.Brick_1X2_DarkGreen'	0%
20	b'4114511.BrickCorner_1X2X2_DarkGray'	6%
21	b'4124091.CornerPlate_1X2X2_DarkGray'	0%
22	b'4125281.BrickCorner1X2X2_DarkGreen'	0%
23	b'4211385.Brick_2X4_MediumStoneGray'	0%
24	b'4211387.Brick_2X2_MediumStoneGray'	93%
25	b'4211397.Plate_2X2_MediumStoneGray'	0%
26	b'4211413.FlatTile_2X2_MediumStoneGray'	0%

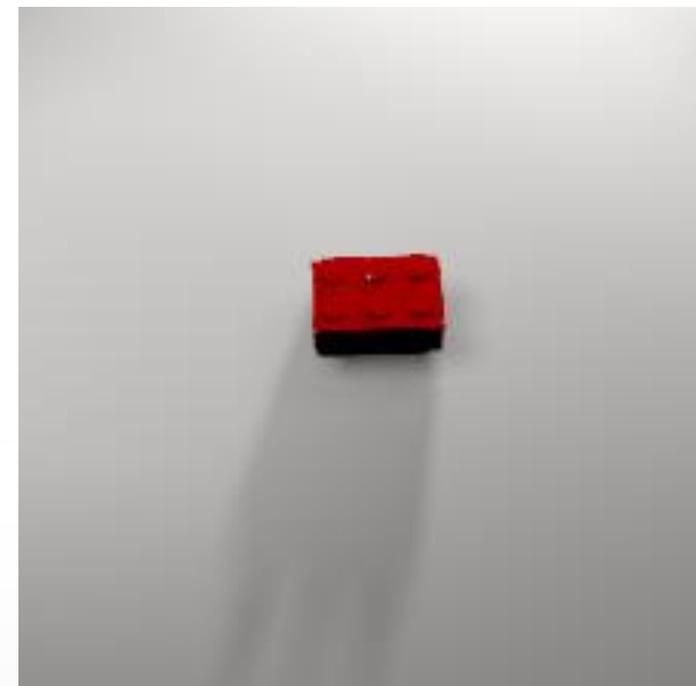
## パラメータ自動チューニング実験2



例示画像



レゴCGデータ



CG画像

# 例示画像の推論結果

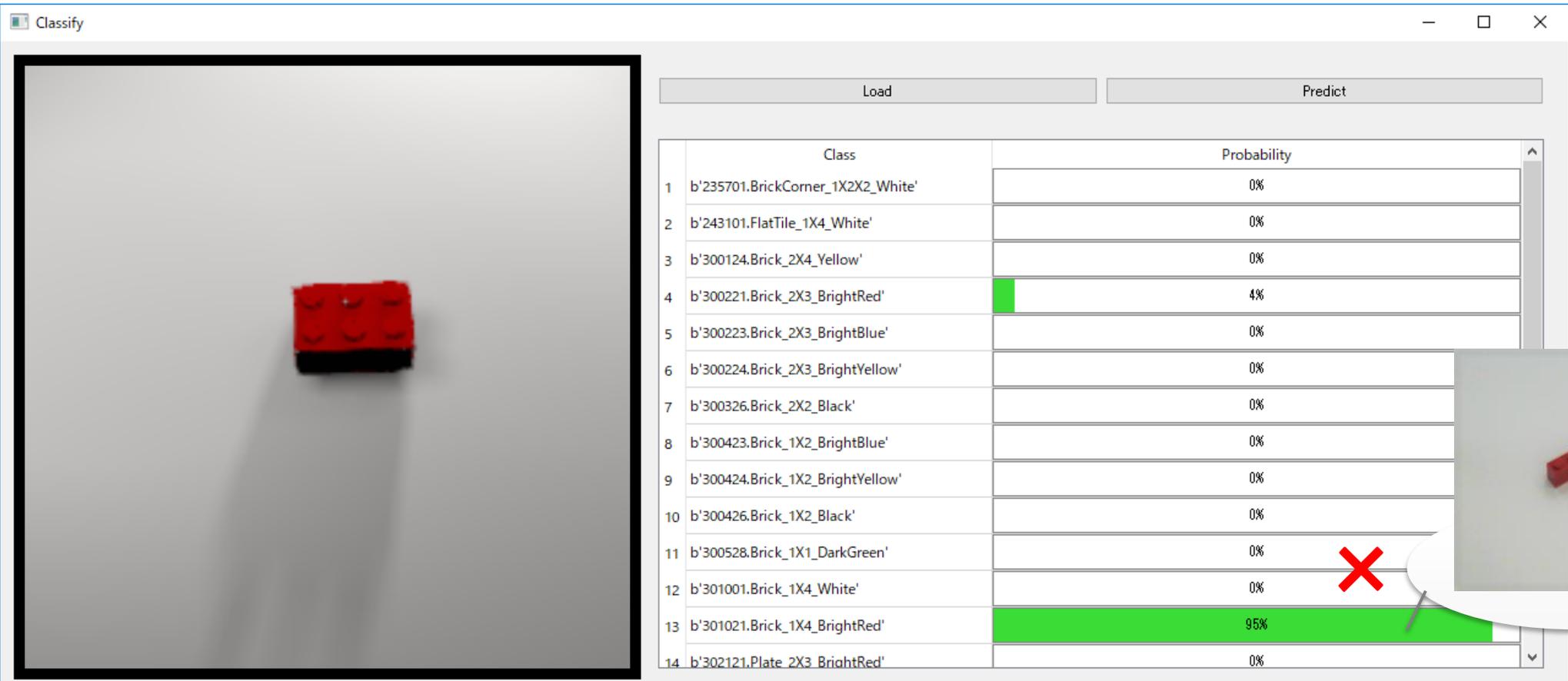
Classify

Load Predict



	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	99%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%

# CG画像の推論結果（自動チューニング前）



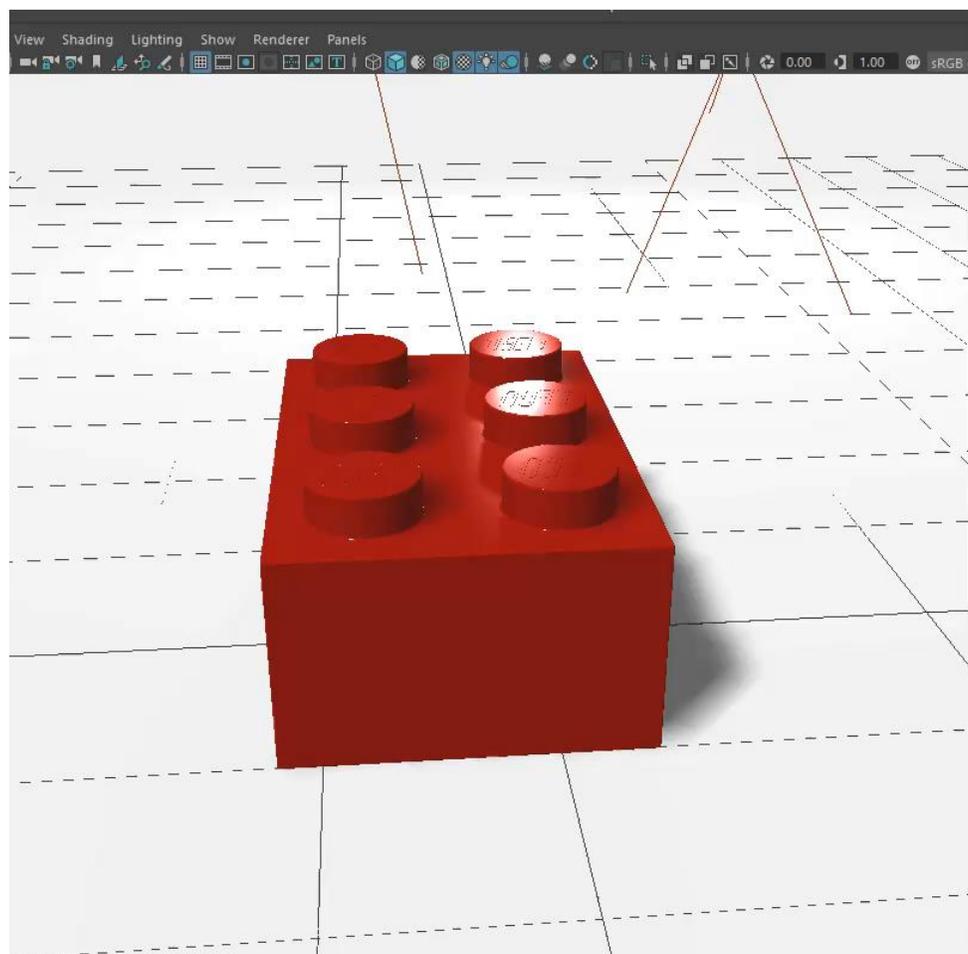
Classify

Load Predict

	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	4%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	95%
14	b'302121.Plate_2X3_BrightRed'	0%

©Silicon Studio Corp., all rights reserved.

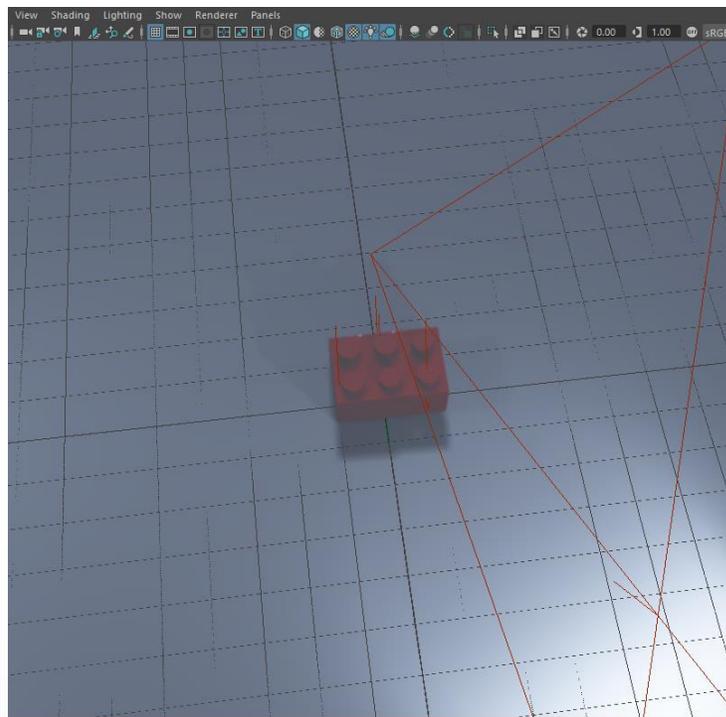
# パラメータ自動チューニングの様子



# パラメータ自動チューニング結果



例示画像



レゴCGデータ

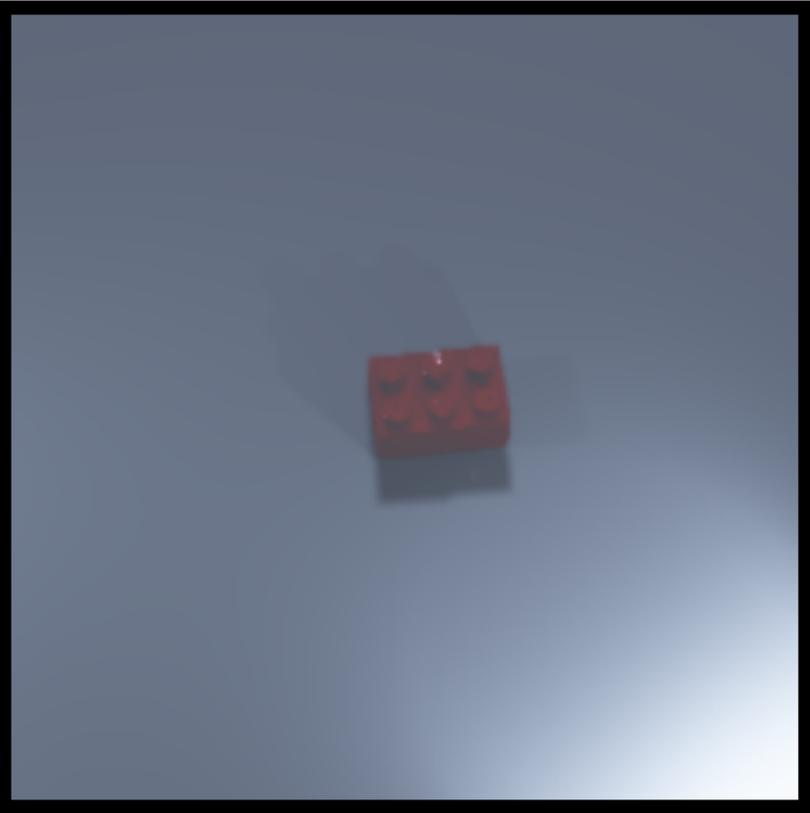


CG画像

# パラメータ自動チューニング後のCG画像推論結果

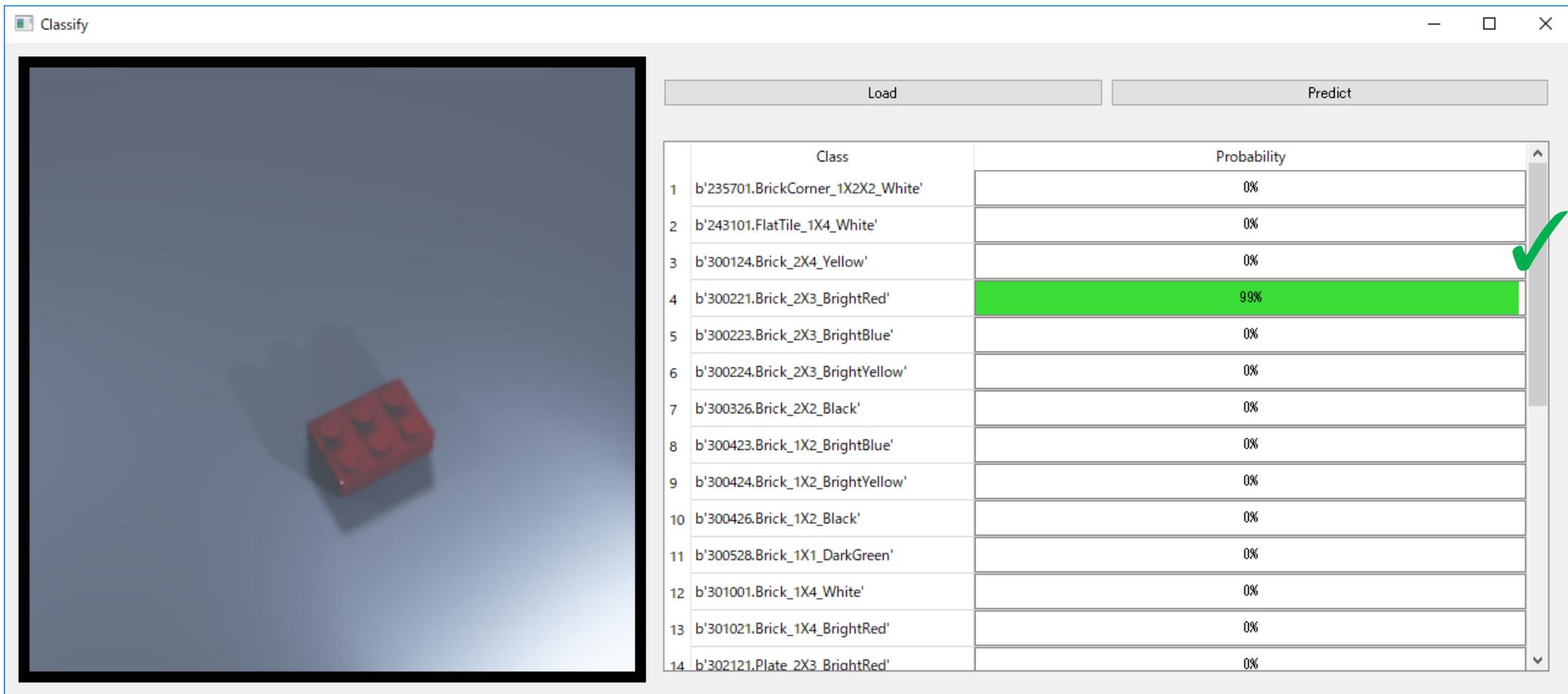
Classify

Load Predict



	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	99%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%

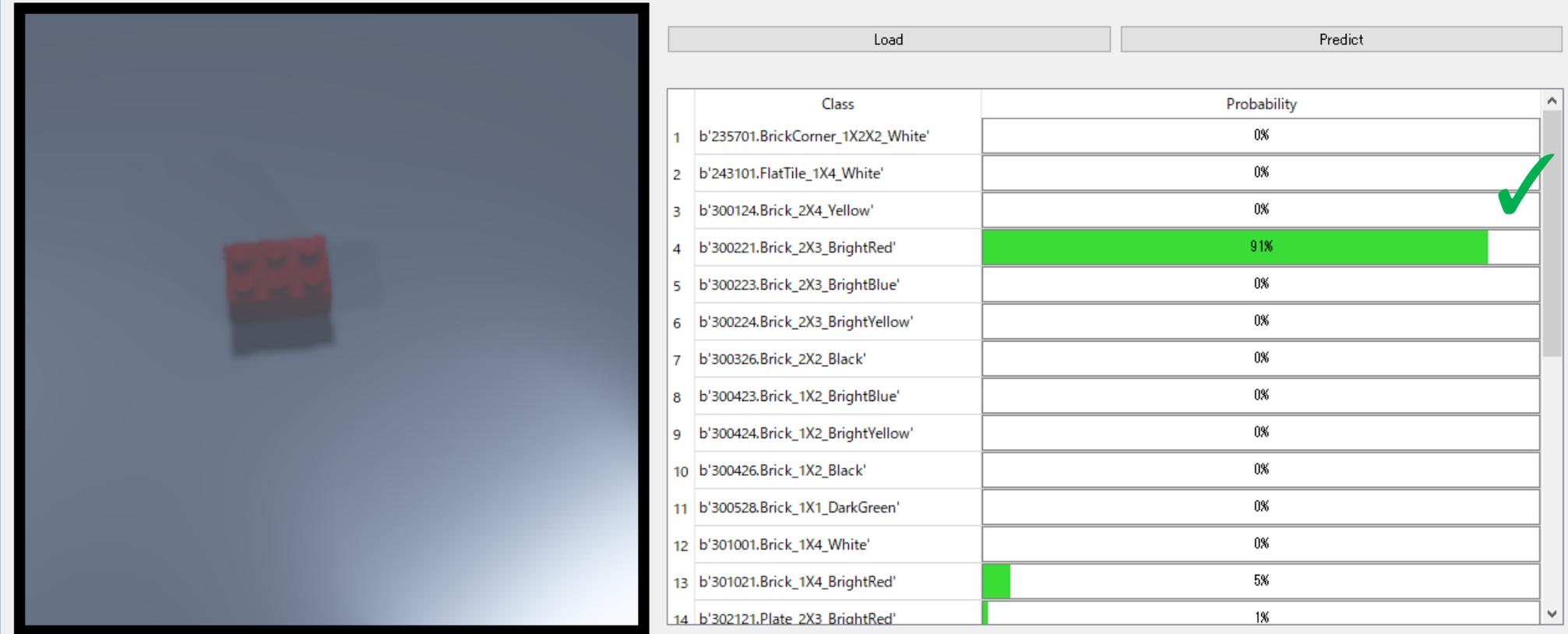
# 自動チューニング後にCGデータ編集



The screenshot shows the 'Classify' window in Silicon Studio. On the left, a 3D model of a red brick is displayed. On the right, a table lists classification results for various brick classes. The table has two columns: 'Class' and 'Probability'. The row for 'b'300221.Brick\_2X3\_BrightRed' is highlighted in green, and a green checkmark is placed to its right, indicating a successful classification.

	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	99%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%

# 自動チューニング後にCGデータ編集



The screenshot shows the 'Classify' window in Silicon Studio. On the left, a 3D model of a red brick is displayed. On the right, a table shows the classification results for 14 different brick classes. The table has two columns: 'Class' and 'Probability'. The class 'b'300221.Brick\_2X3\_BrightRed' is highlighted in green and has a probability of 91%. A green checkmark is placed to the right of this row. The other classes have probabilities of 0% or 5%.

	Class	Probability
1	b'235701.BrickCorner_1X2X2_White'	0%
2	b'243101.FlatTile_1X4_White'	0%
3	b'300124.Brick_2X4_Yellow'	0%
4	b'300221.Brick_2X3_BrightRed'	91%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	5%
14	b'302121.Plate_2X3_BrightRed'	1%

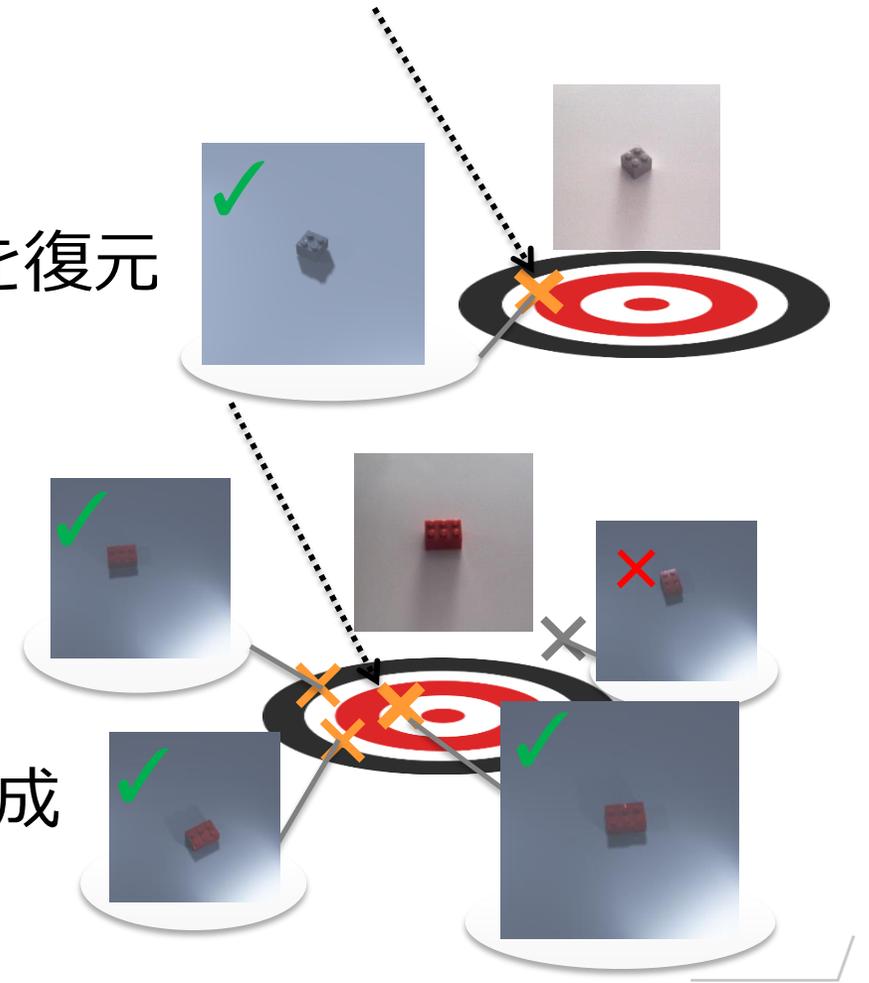
# 自動チューニング後にCGデータ編集

The screenshot shows the 'Classify' window in Silicon Studio. On the left is a 3D render of a red brick. On the right is a 2D image of a green brick. The central table displays classification results for various brick classes. The top row shows a 43% probability for a red brick, while the bottom row shows a 55% probability for a green brick, which is marked with a red 'X' and a callout bubble pointing to the 2D image.

	Class	Probability
4	b'300221.Brick_2X3_BrightRed'	43%
5	b'300223.Brick_2X3_BrightBlue'	0%
6	b'300224.Brick_2X3_BrightYellow'	0%
7	b'300326.Brick_2X2_Black'	0%
8	b'300423.Brick_1X2_BrightBlue'	0%
9	b'300424.Brick_1X2_BrightYellow'	0%
10	b'300426.Brick_1X2_Black'	0%
11	b'300528.Brick_1X1_DarkGreen'	0%
12	b'301001.Brick_1X4_White'	0%
13	b'301021.Brick_1X4_BrightRed'	0%
14	b'302121.Plate_2X3_BrightRed'	0%
15	b'302323.Plate_1X2_BrightBlue'	0%
16	b'302326.Plate_1X2_Black'	0%
17	b'371001.Plate_1X4_White'	0%
18	b'371026.Plate_1X4_Black'	0%
19	b'4107736.Brick_1X2_DarkGreen'	0%
20	b'4114511.BrickCorner_1X2X2_DarkGray'	0%
21	b'4124091.CornerPlate_1X2X2_DarkGray'	0%
22	b'4125281.BrickCorner1X2X2_DarkGreen'	55%
23	b'4211385.Brick_2X4_MediumStoneGray'	0%

## 実験結果

- 例示画像とCG画像の差異
  - おおよそのライト方向/カメラ撮影姿勢を復元
  - 青味を帯びた色調
- CG画像の推論結果
  - 自動チューニングによる改善効果を確認
  - チューニング後編集でバリエーション生成



# もくじ

- 背景と目的
- CG制作のアプローチ
- 課題
- 解決方法
- 実現手段
- 実験と結果
- **まとめ**

## 結論

- 機械学習向けCG画像データの品質改善手法を検討
- CGパラメータの自動チューニング手法を開発
- レゴブロックを使った検証実験を実施
- 機械学習に役立つCG画像を作れる見通しを得た
  - 自動化による合わせ込み品質改善
  - バリエーション生成

## 今後の計画

- レゴブロックでの実験
  - 十分な数の実験
  - CG画像で学習 / 写真で推論
- より実践に近い対象で自動チューニング検証
  - キズ/クラック写真 → プロシージャルテクスチャ
  - 人物画像 → 人体モデルリグ
- パラメータチューニング技術の改善
  - 実写とCG画像との差異の要因把握
  - チューニング処理の高速化