

# KuroNet: Pre-Modern Japanese Kuzushiji Character Recognition with Deep Learning

---

Tarin Clanuwat\* Center for Open Data in the Humanities National  
Institute of Informatics Tokyo, Japan tarin@nii.ac.jp

Alex Lamb\* MILA Universite de Montreal Montreal, Canada  
lambalex@iro.umontreal.ca

Asanobu Kitamoto Center for Open Data in the Humanities National  
Institute of Informatics Tokyo, Japan kitamoto@nii.ac.jp

# 論文概要

---

この論文では以下の点について述べられている

1. 日本語とくずし字の概要
2. KuroNetの概要と詳細
3. 関連研究
4. くずし字認識の課題とアプローチ
5. 実験と今後の課題

# 1.くずし字の概要

## くずし字とは…

- ・約150年ほど前まで使われていた筆記体文字
- ・現代語に比べて文字数が多い
- ・漢字や平仮名のくずれ方が、文脈や人の癖、時代によって異なる



# 1.くずし字の概要

## テキストの特徴

- 文脈でしか判断できないような書き方がある
- ふりがなが存在する
- レイアウトが自由



# 1. 日本語の概要

---

- 漢字は8世紀から使われる
- ひらがな・カタカナは漢字が変形して生まれた
- ひらがなは主に女性が、漢字は主に男性が使用
  
- 日本には明治時代以前に書かれた本が300万冊以上保存
  
- 1900年にひらがなの書き方を統一
  - 現在読める人がほとんどいない
  - 人の手によるデジタル化が困難

# 論文概要

---

1. 日本語とくずし字の概要

 2. KuroNetの概要と詳細

3. 関連研究

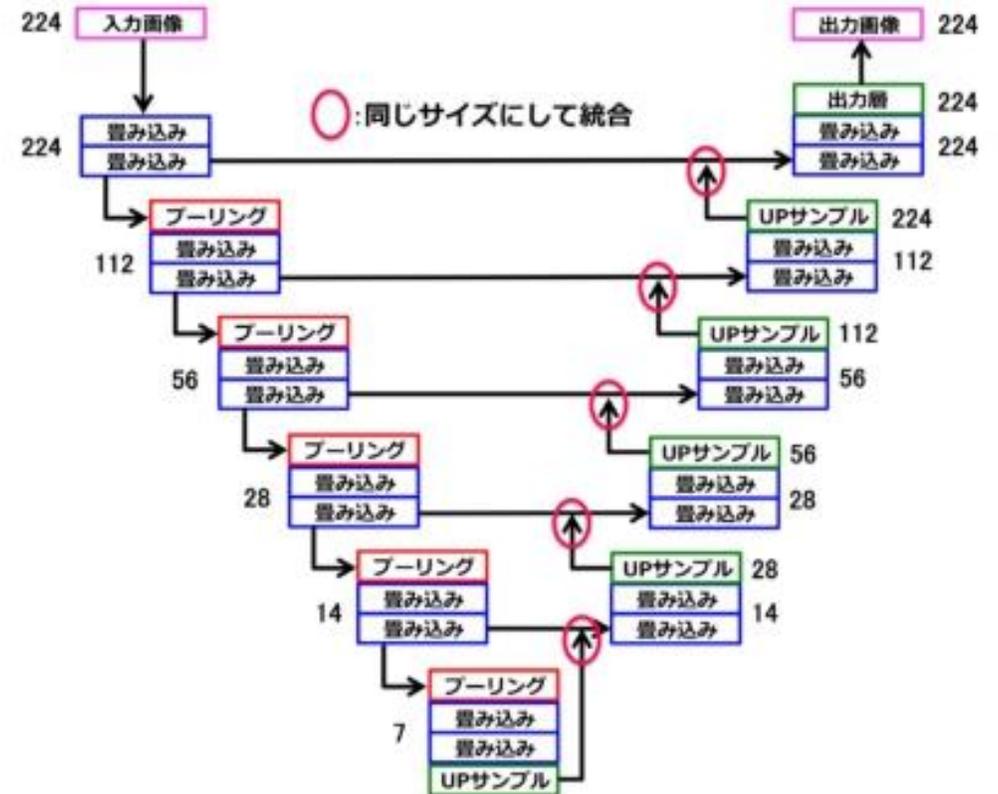
4. くずし字認識の課題とアプローチ

5. 実験と今後の課題

## 2.KuroNetの概要

### KuroNetとは…

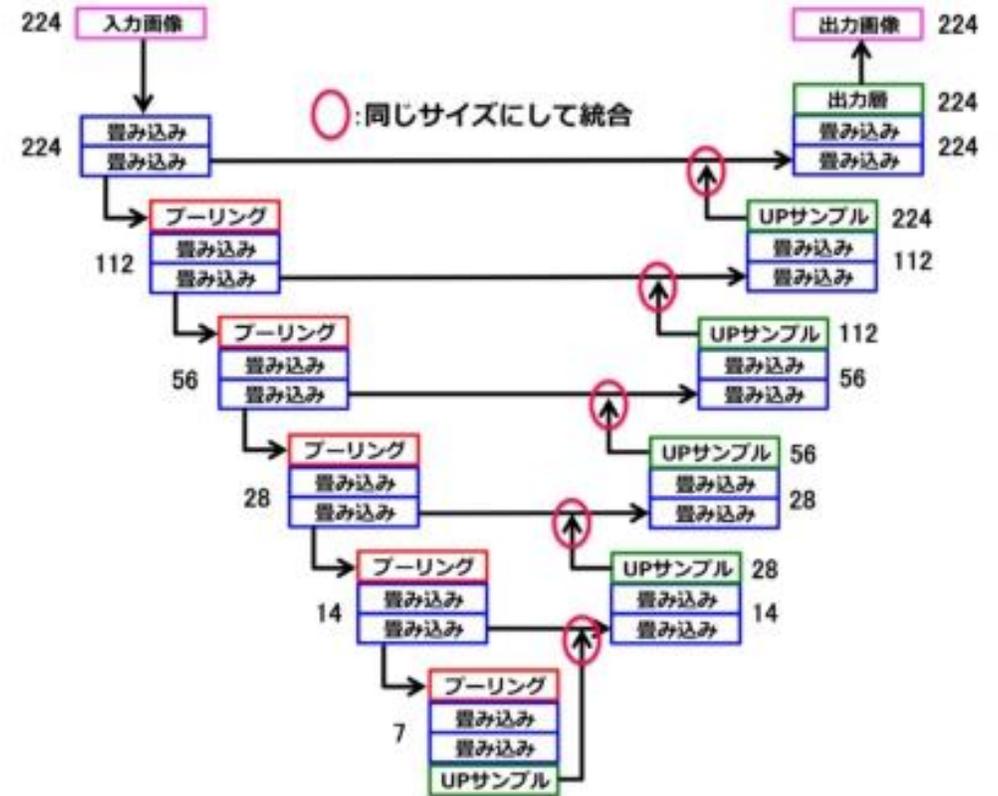
- ページ全体を同時に処理するという考えのもと作り出された、くずし字を認識するための深層学習モデル
- 基本的に事前の処理を必要とせず、文字の位置を使用してトレーニングを行う



<https://lp-tech.net/articles/5Mleh>

## 2.KuroNetの概要

- U-Netアーキテクチャを使用
- U-NetはCNNのひとつ
- 最大の特徴として、  
全結合層が無くUPサンプリングと  
畳み込みによって局所的特徴を  
保持したまま復元可能



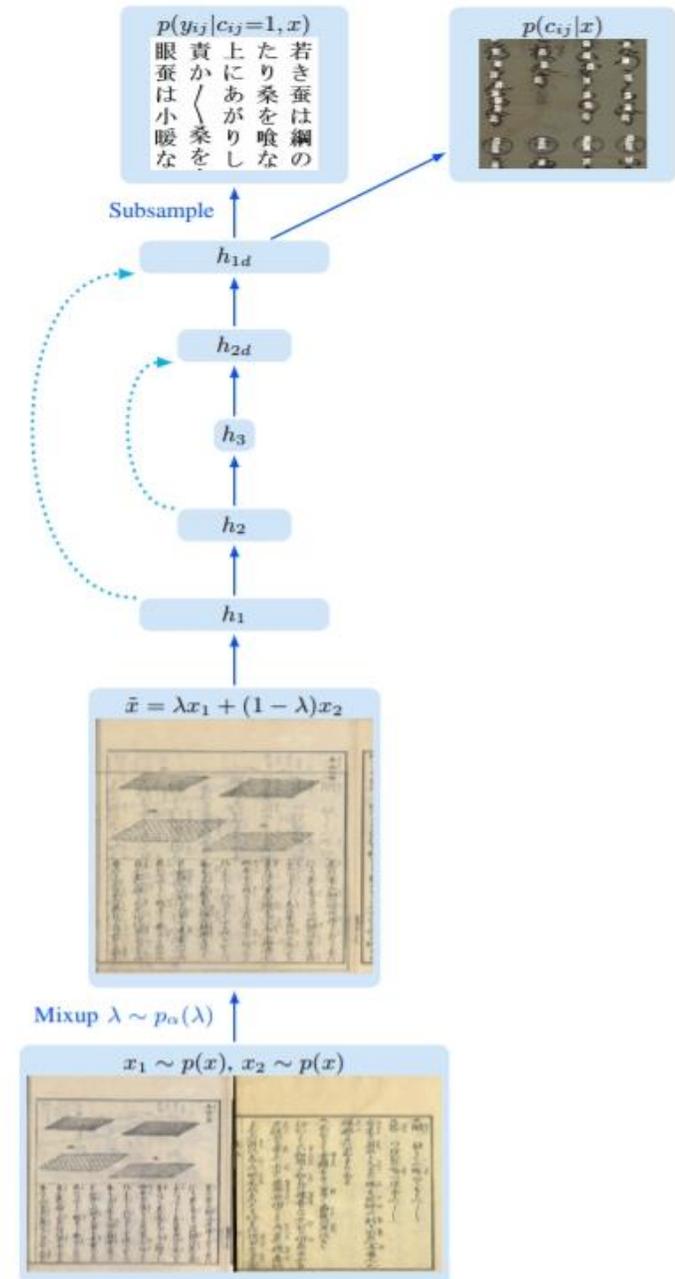
<https://lp-tech.net/articles/5Mleh>

## 2. KuroNetの概要

- 640\*640にリサイズされた画像をU-Netアーキテクチャに通す
- データセットが大きいのので、ベルヌーイ分布を用いて文字領域と背景を分割

↓  
メモリ使用量を大幅削減(Teacher Forcing)

$x \sim p(x)$	: 入力画像
$(i, j)$	: 位置座標
$y_{ij} \sim P(y_{ij} x)$	: 文字位置 (位置座標での多項分布)
$P(c_{ij} x)$	: ベルヌーイ分布
$P(y_{ij} c_{ij} = 1, x)$	: 文字領域内の文字分布

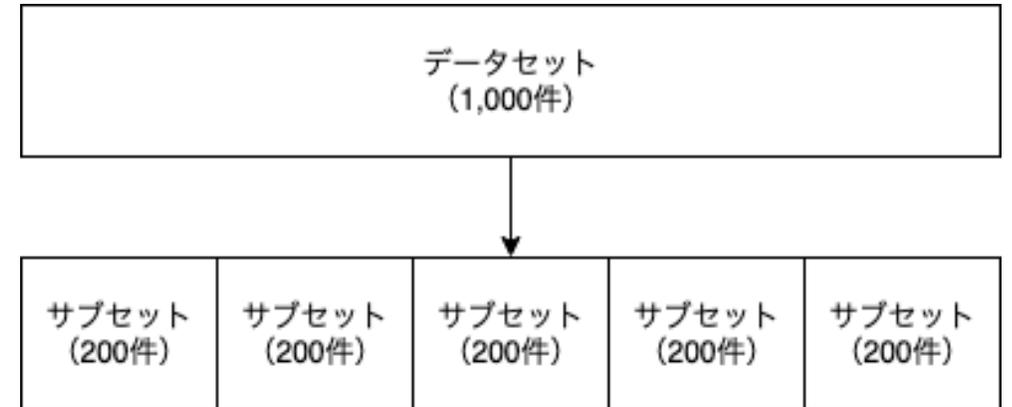


## 2 .KuroNetの詳細

---

### トレーニング

- バッチサイズは 1
  - GPUメモリ内に収まるようにするため
  - 高解像度画像を使用する方が精度向上につながると分かったため
- 通常はサブセットに分割することで、学習時間を短縮する
- エポック（学習回数）は 80



この場合、バッチサイズ（ミニバッチサイズ）は200となる。

<https://qiita.com/kenta1984/items/bad75a37d552510e4682>

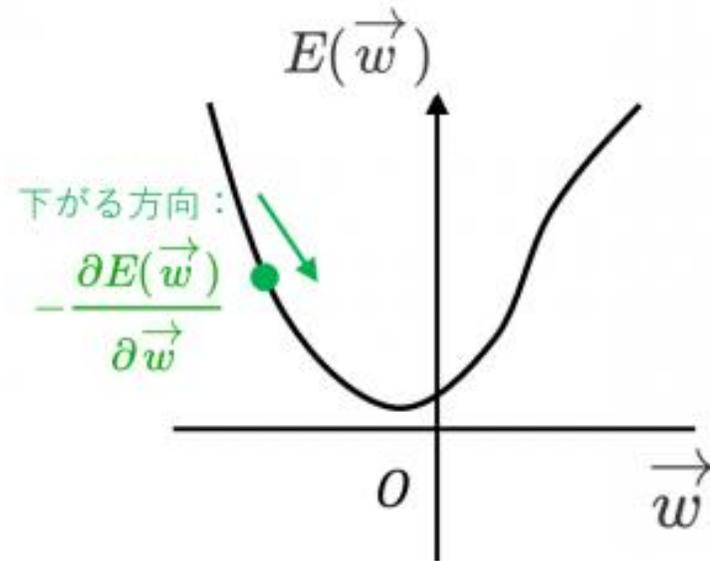
## 2. KuroNetの詳細

### Optimizer(最適化)

- ・ 損失(実値と予測値のずれ)を小さくするパラメーターを探す手法
- ・ トレーニングでは最小になるまで計算を行う

- ・ ベースは勾配降下法
  - ・ ある位置での損失関数の勾配を計算し、勾配の下がる方向に計算を繰り返して最少となる値を探す手法

- ・  $\vec{w}$  : 初期位置



## 2. KuroNetの詳細

---

### Optimizer(最適化)

- AdamOptimizerを使用
  - 勾配の平均と分散を計算し  
モーメントを推定
    - 現状最も使用されてる手法

設定パラメータ

$$\alpha = 0.0001$$

$$\beta_1 = 0.9$$

$$\beta_2 = 0.999$$

$$m_{t+1} = \beta_1 m_t + (1 - \beta_1) \nabla E(\mathbf{w}^t)$$

$$v_{t+1} = \beta_2 v_t + (1 - \beta_2) \nabla E(\mathbf{w}^t)^2$$

$$\hat{m} = \frac{m_{t+1}}{1 - \beta_1^t}$$

$$\hat{v} = \frac{v_{t+1}}{1 - \beta_2^t}$$

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \alpha \frac{\hat{m}}{\sqrt{\hat{v} + \epsilon}}$$

## 2. KuroNetの詳細

---

### 正規化

- ・ 出力を行う前に正規化を行う層を追加することで、出力を抑えたり畳込みの重みを一定に保つことができる
- ・ バッチ正規化(バッチ数を単位とする正規化)の計算式

$$OUTPUT(b, c, x, y) = \gamma_c \frac{INPUT(b, c, x, y) - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c$$

INPUT(b,c,x,y) : 入力(バッチ,チャンネル,x座標,y座標)

OUTPUT(b,c,x,y) : 出力(バッチ,チャンネル,x座標,y座標)

$\mu_c$  : cの平均値、 $\sigma_c$  : cの標準偏差、 $\epsilon$  : 補正、 $\gamma, \beta$  : パラメータ

## 2. KuroNetの詳細

---

### バッチ正規化→グループ正規化

- グループ正規化はチャンネルをいくつかのグループに分けて正規化  
→本論文では16グループに分割
- バッチ正規化はバッチサイズが小さい場合エラーが多くなる  
→グループ正規化ならバッチサイズの影響はないため精度は変わらない

## 2. KuroNetの詳細

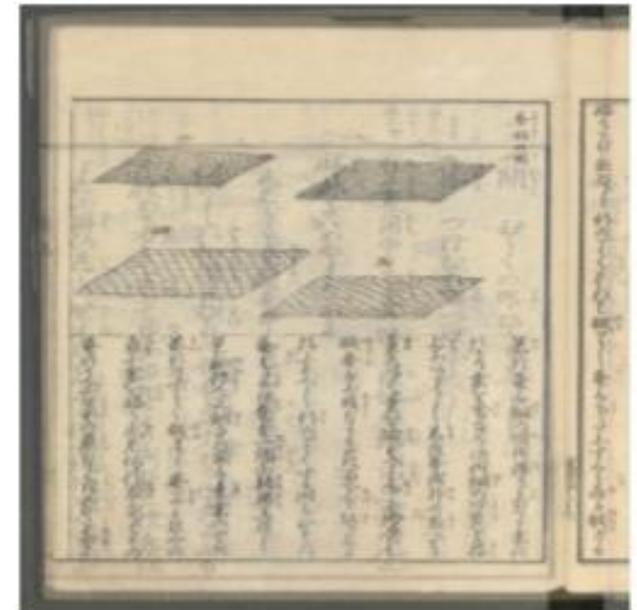
### mixup

- ・ サンプル同士を掛け合わせ、新しいサンプルを作る

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j\end{aligned}$$

$x$ : データ  $y$ : ラベル  $\lambda \in [0,1]$ 、 $\lambda \sim \beta(\alpha, \alpha + 1)$   $\alpha \in (0, \infty)$

- ・ 右図は  $\lambda = 0.3$ の時の合成画像
- ・ 文字認識の手助けとはならないが、学習の際に紙が重なって認識がしづらいページを無視するのに役立つ→結果として精度は向上



# 論文概要

---

1. 日本語とくずし字の概要
2. KuroNetの概要と詳細
-  3. 関連研究
4. くずし字認識の課題とアプローチ
5. 実験と今後の課題

# 3. 関連研究

---

1. イメージを個々のパッチに分割し、各パッチの文字を個別に分類する  
→セグメント化されたデータセットを用意したが、くずし字の文脈的性質から不適切
2. シーケンスモデル  
→くずし字は必ずしもつながりがあるわけではない  
→ページ全体を処理するモデルより予測速度が遅い可能性

# 論文概要

---

1. 日本語とくずし字の概要
2. KuroNetの概要と詳細
3. 関連研究
-  4. くずし字認識の課題とアプローチ
5. 実験と今後の課題

## 4. 課題とアプローチ

---

- 文脈

→KuroNetでは幅広いテキストを使用して曖昧さを回避

- 文字数

→文字領域を抽出することで計算上は対処

- 変体仮名

→一つのクラスに対して複数の書き方が存在するため、各クラスの特徴の分布をとらえる必要がある

## 4. 課題とアプローチ

---

- 注釈

→KuroNetでは無視

- レイアウト

→縦に並んでいない文章やイラストに重なっている文字などが存在  
データセットの補強で対応

# 論文概要

---

1. 日本語とくずし字の概要
2. KuroNetの概要と詳細
3. 関連研究
4. くずし字認識の課題とアプローチ
-  5. 実験と今後の課題

# 5. 実験と今後の課題

---

## 実験 1

- ・ KuroNetを用いた機械学習と文字認識

## 実験の目的

1. KuroNetがどの程度認識できているのかを検証、評価
2. KuroNetが上手く機能しない場合の事例を把握し、今後の課題の見出す

# 5. 実験と今後の課題

---

## 使用するデータ

- National of Japanese Literature(NIJL)が作成したデータセット  
(Center for Open Data in the Humanities(CODH)にて収集されている)
  - 4645文字のクラスと684,165文字以上の画像
  - データセットはバウンディングボックスで構成
- またトレーニング中に見られなかった4つの本もCODHから使用
- 640\*640にリスケールし、イラストや文字が少ないページを省く
  - 誤認識を増やす恐れがあるから

# 5. 実験と今後の課題

---

## 定量的評価 —精度と再現率—

- ・ 精度… (正確に予測された文字数) ÷ (予測文字数の合計)
- ・ 再現率… (正確に予測された文字数) ÷ (ground truth内の文字の合計)

※ground truth…正解データ

- ・ F1スコア… 精度と再現率の調和平均

$$\frac{1}{H} = \frac{1}{2} \left( \frac{1}{p} + \frac{1}{r} \right)$$

H : 調和平均、p : 精度、r : 再現率

# 5. 実験と今後の課題

---

## 結果

- ・ トレーニング中に使ったページと初見の本を比べると、  
トレーニング中に使ったページの方が精度・再現率・F1スコアが高い
- ・ 70%以上F1スコアが0.80を超えている
- ・ F1スコアが0.7以下の本が2冊ある
  - 「brsk00000」… 辞書。稀な単語が多数存在
  - 「200021869」… 料理本。以上に大きな文字が存在
- ・ 類似する漢字を誤検出

# 5. 実験と今後の課題

TABLE I

PERFORMANCE OF KURONET ACROSS RANDOMLY HELD-OUT PAGES FROM DIFFERENT BOOKS.

Book	Book Topic	Year	Number of Classes	Character GT	UNet-Small F1-Score	ResUNet F1-Score	ResUNet+Mixup F1-Score
100249376	Food	1718	401	1127	0.8825	0.9184	<b>0.9256</b>
200021644	Food	1841	785	943	0.7822	0.8885	<b>0.9194</b>
200021712	Food	1786	843	2517	0.8270	0.8661	<b>0.9120</b>
200021851	Medical	1802	430	429	0.8335	0.8389	<b>0.9081</b>
100249371	Food	1852	729	972	0.7866	0.8926	<b>0.8987</b>
hnsd00000	Literature	1838	1972	8518	0.8432	0.8563	<b>0.8915</b>
100249416	Food	1805	469	1021	0.8355	0.8659	<b>0.8892</b>
100249537	Food	1764	826	1288	0.8089	0.8648	<b>0.8857</b>
200014740	Literature	1765	1969	3369	0.8144	0.8521	<b>0.8849</b>
200003076	Literature	1682	1720	5871	0.8092	0.8239	<b>0.8792</b>
100241706	Literature	1834	801	1119	0.8261	0.8315	<b>0.8791</b>
umgy00000	Literature	1832	1737	7746	0.8379	0.8401	<b>0.8757</b>
200021853	Food	1836	595	839	0.8474	0.8563	<b>0.8736</b>
100249476	Food	n/a	644	684	0.7818	0.8513	<b>0.8713</b>
200015779	Literature	1813	1817	5742	0.8291	0.8145	<b>0.8631</b>
200021802	Food	1643	560	1981	0.8084	0.7877	<b>0.8576</b>
200005598	Literature	1790	660	1669	<b>0.8545</b>	0.7794	0.8235
200004148	Literature	1807	2061	3854	0.7582	0.7716	<b>0.8136</b>
200022050	Food	1684	255	526	0.7897	0.7679	<b>0.8122</b>
200021660	Agriculture	1803	1758	3387	0.6541	0.7143	<b>0.7542</b>
200003967	Literature	1878	1119	1278	0.7447	0.7051	<b>0.7535</b>
200021925	Food	1861	693	544	0.6821	0.6481	<b>0.7534</b>
200021763	Food	n/a	704	1189	0.6350	0.7015	<b>0.7495</b>
200021637	Food	n/a	417	422	0.6322	0.7253	<b>0.7343</b>
200014685	Literature	1842	1780	1287	0.6624	0.6751	<b>0.7289</b>
brsk00000	Dictionary	1775	2197	7175	<b>0.7436</b>	0.6309	0.6258
200021869	Food	n/a	330	216	<b>0.6321</b>	0.5471	0.5865

0.8 ↑

0.8 ↓

0.7 ↓

TABLE II

PERFORMANCE OF KURONET AND ABLATIONS ACROSS DIFFERENT METRICS, AGGREGATED OVER MULTIPLE BOOKS. WE REPORT RESULTS FOR BOTH HELD-OUT PAGES (FROM BOOKS SEEN DURING TRAINING) AND HELD-OUT BOOKS (NOT SEEN AT ALL DURING TRAINING).

Book	UNet-Small	ResUNet	ResUNet + Mixup
Held-Out Pages (473 total pages)			
Precision	0.8440	0.8182	<b>0.8683</b>
Recall	0.7901	0.8120	<b>0.8417</b>
F1 Score	0.8162	0.8151	<b>0.8548</b>
Held-Out Books (394 total pages)			
Precision	0.7295	0.7353	<b>0.8155</b>
Recall	0.6839	0.7239	<b>0.7743</b>
F1 Score	0.7060	0.7296	<b>0.7944</b>

# 5. 実験と今後の課題

---

## 実験 2

- ・ アブレーション実験

## 実験の目的

- ・ モデルのどのような側面が重要なのかを理解するため

## 5. 実験と今後の課題

---

### mixupを削除

- ・ 精度・再現率ともに低下  
→mixupは精度向上に効果あり

標準のU-Netを使用、予測文字数を頻度の高い409文字に  
(UNet-Small)

- ・ 精度は上がったが、再現率は低下  
→予測文字数を絞ったことにより、希少な文字を予測することが無いため  
精度が向上

# 5. 実験と今後の課題

TABLE I

PERFORMANCE OF KURONET ACROSS RANDOMLY HELD-OUT PAGES FROM DIFFERENT BOOKS.

Book	Book Topic	Year	Number of Classes	Character GT	UNet-Small F1-Score	ResUNet F1-Score	ResUnet+Mixup F1-Score
100249376	Food	1718	401	1127	0.8825	0.9184	<b>0.9256</b>
200021644	Food	1841	785	943	0.7822	0.8885	<b>0.9194</b>
200021712	Food	1786	843	2517	0.8270	0.8661	<b>0.9120</b>
200021851	Medical	1802	430	429	0.8335	0.8389	<b>0.9081</b>
100249371	Food	1852	729	972	0.7866	0.8926	<b>0.8987</b>
hnsd00000	Literature	1838	1972	8518	0.8432	0.8563	<b>0.8915</b>
100249416	Food	1805	469	1021	0.8355	0.8659	<b>0.8892</b>
100249537	Food	1764	826	1288	0.8089	0.8648	<b>0.8857</b>
200014740	Literature	1765	1969	3369	0.8144	0.8521	<b>0.8849</b>
200003076	Literature	1682	1720	5871	0.8092	0.8239	<b>0.8792</b>
100241706	Literature	1834	801	1119	0.8261	0.8315	<b>0.8791</b>
umgy00000	Literature	1832	1737	7746	0.8379	0.8401	<b>0.8757</b>
200021853	Food	1836	595	839	0.8474	0.8563	<b>0.8736</b>
100249476	Food	n/a	644	684	0.7818	0.8513	<b>0.8713</b>
200015779	Literature	1813	1817	5742	0.8291	0.8145	<b>0.8631</b>
200021802	Food	1643	560	1981	0.8084	0.7877	<b>0.8576</b>
200005598	Literature	1790	660	1669	<b>0.8545</b>	0.7794	0.8235
200004148	Literature	1807	2061	3854	0.7582	0.7716	<b>0.8136</b>
200022050	Food	1684	255	526	0.7897	0.7679	<b>0.8122</b>
200021660	Agriculture	1803	1758	3387	0.6541	0.7143	<b>0.7542</b>
200003967	Literature	1878	1119	1278	0.7447	0.7051	<b>0.7535</b>
200021925	Food	1861	693	544	0.6821	0.6481	<b>0.7534</b>
200021763	Food	n/a	704	1189	0.6350	0.7015	<b>0.7495</b>
200021637	Food	n/a	417	422	0.6322	0.7253	<b>0.7343</b>
200014685	Literature	1842	1780	1287	0.6624	0.6751	<b>0.7289</b>
brsk00000	Dictionary	1775	2197	7175	<b>0.7436</b>	0.6309	0.6258
200021869	Food	n/a	330	216	<b>0.6321</b>	0.5471	0.5865

TABLE II

PERFORMANCE OF KURONET AND ABLATIONS ACROSS DIFFERENT METRICS, AGGREGATED OVER MULTIPLE BOOKS. WE REPORT RESULTS FOR BOTH HELD-OUT PAGES (FROM BOOKS SEEN DURING TRAINING) AND HELD-OUT BOOKS (NOT SEEN AT ALL DURING TRAINING).

Book	UNet-Small	ResUNet	ResUNet + Mixup
Held-Out Pages (473 total pages)			
Precision	0.8440	0.8182	<b>0.8683</b>
Recall	0.7901	0.8120	<b>0.8417</b>
F1 Score	0.8162	0.8151	<b>0.8548</b>
Held-Out Books (394 total pages)			
Precision	0.7295	0.7353	<b>0.8155</b>
Recall	0.6839	0.7239	<b>0.7743</b>
F1 Score	0.7060	0.7296	<b>0.7944</b>

# 5. 実験と今後の課題

---

## 今後の課題

- 大きい文字や複雑なレイアウトの認識
- ふりがなの認識  
→現在はふりがなは無視するように設定されている
- 文字数が多い(変体仮名や漢字)  
→出力前の重みづけは文字ごとに行っているが、これを何らかの方法でグループ化などを行うと、より精度が向上するかもしれない