

画像の追加を許容する Deep Hashing に基づく類似画像検索 Image Retrieval based on Deep Hashing Allowing Image Addition

YE CHENYANG[†] 古賀 久志[‡]

電気通信大学 情報理工学研究科 情報ネットワーク工学専攻

1. はじめに

近年、情報推薦などへの応用を想定し、クエリ画像に似た画像を画像データベースから探す類似画像検索の重要度が高まっており、その中でもハッシュを使った類似画像検索が注目されている。この方式は、画像をハッシュコードでコンパクトに表現した上で類似画像を検索するため、メモリ効率と検索速度の両者が優れていることが長所である。

ハッシュを使った類似画像検索では、ハッシュ関数により検索精度が定まる。深層学習が画像認識で優れた認識性能を達成していることを受けて、データセットに適応的なハッシュ関数を深層学習で学習する Deep Hashing が近年提案されるようになった。Deep Hashing は伝統的なハッシュ関数より高い検索精度を達成する。

しかし、現状の多くの Deep Hashing では、テスト画像と学習画像が同一データセットに所属することを前提としている。つまり、クエリが学習済みのデータセットに属する条件で高い検索精度を保証する。しかし、現実の類似検索システムでは学習していないデータセットからの未知の画像がクエリとなることが当然起こりえて、その際に検索精度が低下することが懸念される。

この状況へ対処することを目的として、本稿では1つのデータセットで学習済みの Deep Hashing に学習画像を追加することを可能とするフレームワークである UIDH (Unsupervised Incremental Deep Hashing) を提案する。UIDH では最初に教示ありで初期データセット 1 に対して高い精度を達成するハッシュコードを生成する。その後、追加されたデータセット 2 に対してインクリメンタル学習を用いて非教示でハッシュコードを生成する。とくに、教示ありのハッシュコードを上位ビットと非教示のハッシュコードを下位ビットとして連結し、上位ビットによりデータセット 1 に対して高い検索精度を維持し、下位ビットでデータセット 2 に対する類似検索もカバーする。このようにして1つの検索システムで初期データセットと追加データセットを同時に取り扱うことを可能にする。

下位ビットを非教示にした理由は、正解クラスを取得するオーバーヘッドなしで画像追加を可能にするためである。さらに下位ビットを非教示にしたことで、初期データセット 1 に対するハッシュコードの種類 (多様性) が増加することで、類似画像検索結果が人間の直感とより合致するという興味深い現象が確認された。

本稿の構成を以下に述べる。2章で我々の研究に関連する既存 Deep Hashing 手法を紹介し、3章でそれらの課題を踏まえて研究目的を述べる。4章でこの課題を解決する提案手法を記述する。5章で提案手法を実験的に評価する。6章がまとめである。

2. 関連研究

この章では今まで重要な既存教示の深層ハッシュ方法を紹介する。2.1章では教示深層ハッシュ手法の分類と近年の発展を述べる。2.2章には今回の研究に注目した離散的な更新法によって非対称な深層ハッシュ手法を述べる。2.3章に 2.2章の深層ハッシュに基づく追加データ許容な学習手法を述べる。

2.1 伝統的な教示 Deep Hashing

深層ニューラルネットワークがコンピュータビジョン領域の成果が画像ハッシュ検索問題に応用されていた、教示の深層ハッシュでは、三つの画像間の類似性を保つ、ペアラベル画像の類似性を保つと非対称深層ハッシング法に分けている。三つの画像間の類似性を保つでは同じラベル画像のハッシュ距離は違うラベル画像距離より近いによってネットとハッシュ関数の学習をする。ペアラベル画像手法の代表では DSH(Deep Supervised Hashing)[1]では。この手法で、二つの画像ラベルが同じかどうかだけを注目し、類似行列を作り出して学習易くなる。以上の手法ではハッシュコードを連続の値に変換し、ネットの学習を行い、その後閾値を用いて連続値をハッシュコードに変換する。そして、近年 DSDH(Deep Supervised Discrete Hashing)[2]という深層ハッシュネット構造が出て直接に離散的なハッシュコードを学習することができるに従って、更に精度が上がる。その影響で、非対称深層ハッシング法 ADSh と DIHN が出てくる。

2.2 教示非対称的な Deep Hashing

非対称深層ハッシング法 ADSh(Asymmetric Deep Supervised Hashing)[3]がデータベースの一部を学習データポイントとして学習の途中にネットのパラメータを更新するに従って、データベース画像のハッシュコードを直接に離散的な手法で学習する。そのため、伝統的な深層ハッシュと比べて学習時間とコストが下がり、実験結果によって精度も優れている。

具体的に ADSh ネットパラメータ θ の更新とデータベースハッシュコード B の学習が交替で行う。ネットを更新するとき、ハッシュコード行列 B を固定して類似性を示す損失関数でネットワークのパラメータ θ を更新する。類似性を示すため、ハッシュ距離を用いて距離の小さい画像の類似性が高くなり、ハッシュコードの長さ K と類似行列 S の掛け算とハッシュコードの内積が類似性のような論理がある、下記のような損失関数の最小化を目指す：

$$\sum_{i \in \Psi} \sum_{j \in \Psi} [b_i^T \tanh(f(d_j; \theta)) - KS_{ij}]^2 \quad (1)$$

データベース画像の集合は Ψ に表し、その中で取る学習データは ψ に示す。 b_i はデータベースハッシュコード B の i 行目、画像 i のハッシュコードとする。 d_j は学習データ画像 j である。ネットワークパラメタを更新するため、 $\tanh(\cdot)$ で出力は-1から+1までの値にする。そして、同じ画像がネットの出力とハッシュコードを一致するのは望ましいので、新しい制約を加えて最後の ADSH の損失関数を得る：

$$\min_{\theta, B} J(\theta, B) = \sum_{i \in \Psi} \sum_{j \in \psi} [b_i^T \tanh(f(d_j; \theta)) - KS_{ij}]^2 \quad (2)$$

$$+ \gamma \sum_{j \in \psi} [b_j - \tanh(f(d_j; \theta))]^2$$

パラメタ θ では損失関数から後方伝搬処理の最急降下法で更新する。パラメタ θ 更新した後、ハッシュコード行列 B を離散的に学習することを行う。ハッシュコード行列 $B(b_i \in B)$ が離散的な最適解を求める手法 DCC[4]によって、ADSH の損失関数式(2)から変換計算で得る：

$$P = -2kUS^T - 2\lambda\tilde{U}_l$$

$$B_{l*} = -\text{sign}(2\hat{B}_l^T \hat{U}_l U_{l*}^T + P_{l*}^T) \quad (3)$$

ここで B_{l*} 、 U_{l*}^T と P_{l*}^T では行列の l 列目とする。 U は B の拡大行列である、更新のは列目を単位として行う。最後に、 $-\text{sign}(\cdot)$ で離散的なハッシュコードを学習する。

2.3 教示でデータ追加を許容する Deep Hashing

ADSH のネット構造を従って DIHN(Deep Incremental Hashing Network for Efficient Image Retrieval)[5]というインクリメンタルハッシュ学習が出てくる。DIHN が一つのデータセットで種類によって初期データと追加データを分け、初期データに学習済みのネットが追加データにも対応できることを目的として ADSH ネットに学習した後のネットとハッシュコード行列を古いと追加データの和集合にもう一回インクリメンタル学習をする。ネットワークパラメタを更新する場合は、データベース画像の集合は追加データと初期データの和集合として学習画像がその両方データの一部となる。そして、ADSH と同じ方法で類似行列を作り出し、ネットのパラメタとハッシュコード行列を更新する。実験結果によって、一つのデータセットでインクリメンタル学習して ADSH とほとんど同じ程度の結果を得る。

3. 研究目的

2章で述べたように、既存の教示あり Deep Hashing のほとんどは高い検索性能を達成するものの、テスト画像が学習画像と同一データセットに属することを前提条件とする。その結果、未知データセットからのテスト画像に対しては検索精度が低下することが懸念される。そこで、本研究では1つのデータセットで学習済みの Deep Hashing に学習画像を追加することを可能とする仕組み UIDH (Unsupervised Incremental Deep Hashing)を実現することを研究目的とする。既存手法である DINH も学習画像の追加が可能であるが、本研究は(1)非教示で学習画像を追加できる、(2)DINH では単一データセットの一部クラスを追加画像とするのに対して、UIDH では学習済み画像と異なるデータセットに属する画像を追加するという2点が異なる。

また、本研究では ADSH が高い mAP(mean Average Precision)を達成する一方で、非常に少ない種類のハッシュコードしか生成しないため、クエリ画像が変わっても検索結果が変わらないという欠点を持つことを指摘する。図1は、Nuswide データセットで異なる2枚の画像をクエリ(左上隅の画像)として ADSH が生成したハッシュコードで上位29枚の類似画像を求めた検索結果である。クエリ画像が全く異なるにも関わらず、検索結果が全く変わらないという不自然な結果になっている。原因を調べたところ、ADSH では10万枚の画像に対するハッシュコードの種類数が1500未満と非常に少なくハッシュコードの多様性に問題があることがわかった。



図1 従来手法 ADSH の画像検索 top29

UIDH のように教示ありのハッシュコードと非教示にハッシュコードを連結することで、ハッシュコードの多様性を高めることで ADSH の問題点を改善することを示すことも本研究の目的である。

4. 提案手法

本章では、学習済み画像とは別のデータセットからの画像を学習画像として追加できるフレームワーク UIDH を提案する。UIDH では、最初に初期データセット1を学習画像として2K bit のコードを出力するハッシュ関数 h_1 を教示有で学習する K はハッシュコードの長さである。 h_1 の上位 K ビット出力する関数を hu_1 、下位 K ビットを出力する関数を h_1 と呼ぶことにする。

その後、新データセット2が現れた時には、データセット2を学習画像として K bit のコードを出力するハッシュ関数 h_2 を学習する。ここではインクリメンタル学習を行っており、データセット1に属する画像 d について $h_2(d)$ が $h_1(d)$ からかけ離れないよう工夫している。新データセットに関しては非教示でクラスラベルを使わないことで学習画像を準備する手間を軽減することを狙っている。

最終的な画像 d に対するハッシュ値 $h(d)$ は d が属するデータセットに依らず、下式のように hu_1 のハッシュコードを上位ビット、 h_2 のハッシュコードを下位ビットとして連結して出力する。

$$h(d) = (h_{1u}(d), h_2(d))$$

d がデータセット 1 に属する場合は上位ハッシュコード $h1_u(d)$ に高い検索精度を維持する。 d が新データセット 2 に属する場合は下位ハッシュコードで非教示レベルの検索精度を達成する。

さらにデータセット 1 に属する画像群に対しては、下位ビット $h2(d)$ によりハッシュコードの種類が増えて多様性が増す。これにより 3 章で述べたクエリを変えても検索結果が変わらないという問題点を改善することが期待できる。

以降では 4.1 章で初期データセット 1 に対するハッシュ関数の学習方法を説明し、4.2 章で追加データセット 2 に対するハッシュ関数の学習方法を記述する。

UIDH step2

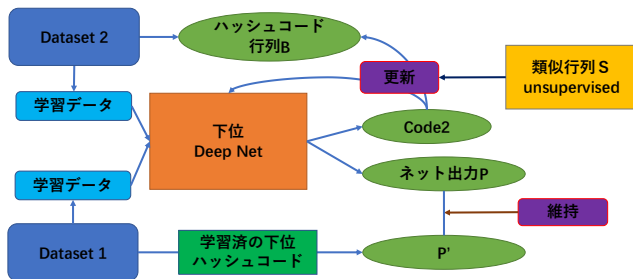


図 2 UIDH step2 に対するネットワーク

4.1 Step 1: 初期データセットに対するハッシュ学習

提案手法のステップ 1 は初期データセットに対する学習である。ここでは高い検索精度を達成するため既存の教示手法を利用する。具体的には ADSSH (Asymmetric Deep Supervised Hashing) を利用した。Step 1 では、以下の 2 種類のバリエーションを考案した。1 は 2K ビットすべてを教示ありで生成しており、mAP のような検索性能を高くしたい時に使うことを想定する。2 は mAP の高さよりもハッシュコードの多様性を重視したい時に使うことを想定している。

1. 2K bit すべてを ADSSH で生成
2. 上位 K ビットを ADSSH で生成し、下位 K ビットを非教示で生成する。具体的には下位ビットは非教示版の ADSSH を用いて生成する。

2 の非教示版の ADSSH は、通常の ADSSH と較べて類似行列 S を非教示でクラスラベル情報なしに構築することのみが異なる。類似行列 S は、非教示の深層分類ネットを使って初期データセットの各データに擬似ラベルを付与し、擬似ラベルから作成する。初期データセットはクラスラベル情報も取得可能で、性質がよくわかっているという前提の下、データセットに応じて異なる深層分類ネットワークを用いる。本研究で使用した 2 つのデータセット *Cifar10* と *Nuswide-tc21* に対してどのように擬似ラベルを付与するかを述べる。

Cifar10 に対しては近年提案された非教示深層分類ネットワーク SCAN[6] で擬似ラベルを作り出す。SCAN ではまず pretext task により特徴表現を学習し、各画像に対する近傍画像を得る。次に近傍画像と原画像が同一クラスになるような分類損失関数を設けて学習し、クラスタリング結

果を得る。類似行列 S はクラスタリング結果より生成される。*Nuswide-tc21* に対しては、非教示 Deep Hashing である SSDH[7] と同様に類似行列を作り出す。SSDH ではデータセットの画像特徴ベクトルのコサイン距離に基づいて、2 枚の画像が -1 (非類似)、0 (どちらでもない)、1 (類似) と判定する。これにより、3 値の類似行列が得られる。しかし、ADSSH ではもともと教示ありの手法なので、類似行列が -1 (非類似)、1 (類似) の 2 値である必要がある。そこで、類似行列の値が 0 なる要素は全て -1 に変換した。

類似行列 S を作った後のハッシュ関数は以下の損失関数を最小化することで学習する。画像 i に対するハッシュコード bi と画像 j に対するニューラルネットワークの出力の内積が S_{ij} の K 倍に近づくようにしている。

$$\min_{\theta, B} J(\theta, B) = \sum_{i \in \Phi} \sum_{j \in \Phi} [b_i^T \tanh(f(d_j; \theta)) - K S_{ij}]^2 + \gamma \sum_{j \in \Phi} [b_j - \tanh(f(d_j; \theta))]^2 \quad (4)$$

4.2 Step 2: 追加データセットに対するハッシュ学習

ステップ 2 では、追加されたデータセット 2 に対して K ビットのコードを生成するハッシュ関数 $h2$ を非教示で学習する。 $h2$ は最終的なハッシュコードの下位 K ビットに対応する。 $h2$ を学習する際は、Step 1 で学習した下位 K ビットにハッシュ関数 $h1$ のデータセット 1 に対する検索精度の高さを維持するためにインクリメンタル学習を行う。具体的には初期データセットの画像 d に対するハッシュ値 $h2(d)$ がステップ 1 のハッシュコードの下位ビット $h1(d)$ に近くなることを促す再生(Replay)項

$$\min_{p'} \sum_{j \in \Psi} |p_j - p'_j|_H^2 \quad (5)$$

を損失関数に追加する。Replay 項はインクリメンタル学習 iCaRL[8] からインスピレーションを得た。iCaRL では追加データを学習するとき、一部の初期データを学習データに加えて追加データと一緒に学習する。式(5)で p_j は初期データセットに含まれる画像 j のステップ 1 で学習した下位ネットの出力であり、 p'_j が画像 j のステップ 2 で学習したネットの出力である。実際には式(5)を変形した式(6)を損失関数に加える。式(6)では画像 j に対する Step2 のネットワークの出力 $f(d_j; \theta)$ を $h1(j)$ に同一にすることを旨とする損失関数である。

$$\sum_{j \in \Psi} [b_j^T \tanh(f(d_j; \theta)) - K]^2 \quad (6)$$

再生項に非教示版 ADSSH の損失関数とハッシュコードの $\{1, -1\}$ の分布をバランスさせる項を追加して、提案手法 UIDH におけるステップ 2 でのニューラルネットワークに対する最終的な損失関数を得る (式(7))。式(7)の類似行列 S は非教示で生成する。 S は本来追加されるデータセットによらないユニバーサルなものであることが望ましいが、

現状では 4.1 章と同様に追加されるデータセットに応じて生成方法を変えている,

$$\begin{aligned} \min_{\theta, B_{\Phi}} J(\theta, B_{\Phi}) = & \sum_{i \in \Phi} \sum_{j \in \Phi} [b_i^T \tanh(f(d_j; \theta)) - K S_{ij}]^2 \\ & + \gamma \sum_{j \in \Phi} [b_j - \tanh(f(d_j; \theta))]^2 \\ & + \mu \sum_{j \in \Phi} [\tanh(f(d_j; \theta)) \mathbf{1}]^2 \\ & + \eta \sum_{j \in \Psi} [b_j^T \tanh(f(d_j; \theta)) - K]^2 \end{aligned} \quad (7)$$

ネットのパラメータ θ を更新するとき、直接に損失関数から後方伝搬処理する。データベースのハッシュコード B_{Φ} を更新するときでは新しい項がハッシュコード B_{Φ} に影響を与えていないため、式(3)と同じように更新する。

5. 実験

今回の研究では、Cifar10 と Nuswide 二つのデータセットに実験をした。システムは Ubuntu 16.04.7 で、GPU は GeForce GTX 1080 Ti である。

5.1 データセット

Cifar10 では 60000 枚の画像を含む単一ラベルデータセットである。オブジェクトのクラス数は 10 である。その中で、各種類を 1000 枚画像を取ってテスト画像として、他の 50,000 枚画像をデータベース画像とする。

Nuswide では 269,648 枚の画像を含むマルチラベルデータセットである。81 のクラス概念があり、この中で 21 によく用いているクラスを取って 100,000 枚画像のデータベースをする。各クラスを 100 枚画像を保持して 2100 枚画像をテスト画像とする。

5.2 パラメータと初期化

ADSH の実験はその論文と同じパラメータでネット構造を AlexNet として 48 ビットのハッシュコードを生成する。

提案手法 UIDH では上位と下位ネットは AlexNet で、最後の全結合層にもう一つの全結合層を連結し、24 ビットのハッシュコードを生成して上位と下位ハッシュコードを加えて 48 ビットをシステムの出力とする。パラメータでは ADSH と同じ γ が 200 で、 μ を 50 とする。学習率は 0.0001、ネットパラメータ θ を三回学習して、一回 B の学習を行う。Epoch は 50 である。

5.3 MAP による検索精度の評価

こちらの実験では、ハッシュコードが同じ 48 ビットで ADSH と UIDH への検索精度 MAP を比較する。提案手法 UIDH には step1 の下位ビットを教示で生成するバージョンと非教示で生成するバージョンがある。前者と UIDH(S) 後者を UIDH(U) と記す。以下の実験で精度を計算するとき、Cifar10 を top50000 を取り、Nuswide で top5000 を取る。赤い文字では初期データセットを示す。

Cifar10 をデータセット 1 Nuswide をデータセット 2 として平均精度の実験結果は表 1 のように：

表 1 MAP(初期データセット Cifar10)

method	Cifar10	Nuswide-tc21
ADSH 48bits	0.9125	0.5158
UIDH (S) 48bits	0.9029	0.7179
UIDH (U) 48bits	0.8809	0.7187

ここで、UIDH の学習では精度が一番いい結果のパラメータを用いてここで $\eta = 0.1$ である。

UIDH(S)の結果から見ると、Cifar10 を初期データとして提案手法と比較手法 ADSH 比べて Cifar10 に対して精度が 1%ぐらいの下がりがあるけれど、追加データ Nuswide での精度を 20%ぐらい上がる。UIDH(U)の結果では、ステップ 1 に下位ネットが誤りの情報を用いて学習するため、初期データの精度が若干下がる。追加データに対して UIDH(S)と同じ程度の結果が出る。

Nuswide をデータセット 1 で Cifar10 をデータセット 2 としての平均精度を表 2 のように示す：

表 2 MAP(初期データセット Nuswide-tc21)

method	Nuswide-tc21	Cifar10
ADSH 48bits	0.9001	0.1662
UIDH (S) 48bits	0.8701	0.5926
UIDH (U) 48bits	0.8709	0.6072

ここでパラメータ $\eta = 0.3$ である。UIDH(S)では、Nuswide を初期データとして提案手法は 3%ぐらい精度が下がるけれど、追加データ Cifar10 への精度を 42%ぐらい高める。UIDH(U)で、提案手法が UIDH(S)と同じ程度結果をえる。

平均精度結果から見ると、UIDH ネット構造が一つのシステムで初期データの精度をある程度に維持しつつ追加データの精度を上げるハッシュコードを生成できる

5.4 topk 画像の比較

Nuswide をデータセット 1 として、図 1 と同じ、2 枚の画像をクエリ (左上隅の画像) として UIDH(U)でハッシュ距離に近い 29 枚の画像を図 3 で表す。パラメータ $\eta = 0.3$ である。図 1 と比べてクエリによって検索結果から見ると、同じクエリ画像に対して提案手法 UIDH が違う検索結果を表す。とくに、クエリ画像 1 と 2 で夕陽と船が一番はっきり表現された特徴けれど、マルチラベルのため従来手法 ADSH の検索結果は別の特徴を注目する。それはハッシュコードが違う画像の細かい特徴を取ることができないため。逆に提案手法 UIDH の検索は夕陽と船の色あるいは形が更に似ているもの検索出て来た。その原因を調べるため、実際にハッシュコードを見て ADSH で二つのクエリ画像は同じハッシュコードを生成された、つまりハッシュコードの種類が少ないくて、ラベルに依存過ぎることが起こる。



図 3 UIDH の画像検索 top29

こちらで比較手法 ADSH と提案手法 UIDH にハッシュコードの種類を計算する、結果は図 4 に表す。結果から見ると、データセット 1 がいずれにも提案手法 UIDH がハッシュコードの種類を高めることができる。そして、ステップ 1 下位ネットが非教示の手法で学習したら、教示の手法よりハッシュコードの種類が更に上げられる。画像検索に対して有利になる。提案手法の非教示の追加データ学習過程で初期データのラベルの依存性のある程度に改善することができる。

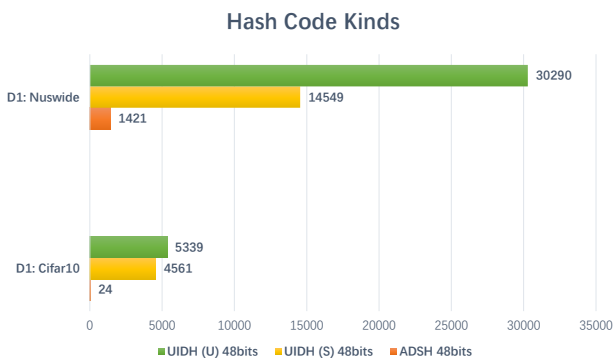


図 4 ハッシュコードの種類

5.5 Replay の作用と感度

こちらの実験では Nuswide をデータセット 1 として、提案されたインクリメンタル学習方法 Replay の作用を表 3 のように示す：

表 3 Replay 項パラメータを変える MAP(平均精度)

パラメータ	Nuswide-tc21	Cifar10
$\eta = 0.0$	0.8334	0.5880
$\eta = 0.3$	0.8701	0.5926
$\eta = 0.5$	0.8663	0.6047

ここで提案手法 UIDH の損失関数に Replay 項のパラメータを変えた結果である。結果によって、追加データでの精度を同じ程度けれど、Replay 項がある場合は初期データの精度を 3% ぐらいに上げられる。

こちらの実験でインクリメンタル学習方法 Replay 項は確かに初期データに精度を維持しすることを貢献がある。

6. おわりに

本研究では、データセットの追加を許容する Deep Hashing UIDH を提案した。提案手法では最初に初期データセットに対するハッシュ関数 h_1 を教示ありで学習し、追加データセットに対するハッシュ関数 h_2 を非教示でインクリメンタルに学習する。最終的なハッシュコードは h_1 の上位ビットに h_2 のハッシュコードの下位ビットを連結したものになる。 h_1 の上位ビットで初期データに対する検索精度を維持しつつ、 h_2 のハッシュコードで追加データに対する検索精度を高める。こうして 1 つのニューラルネットで 2 種類のデータセットに対応する。さらに UIDH は従来手法 ADSH と比べ、ハッシュコードの多様性を高めることで、人間の直感に適合した類似画像検索結果を得られる。

しかし、現状ではデータセットに依って異なるネットワークを使って非教示のハッシュコードを生成している。異なるデータセットに対して統一的なやり方で類似行列を作り出すのが今後の課題である。そして、多様性の高いハッシュコードを得るために、ハッシュコードの種類を増やす損失関数を設計することも今後の課題である。AutoEncoder のような画像特徴を維持す機構の導入が有望であると考えている。

謝辞

本研究は JSPS 科研費 JP21K11901 の助成を受けた。

参考文献

- [1] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. "Deep supervised hashing for fast image retrieval". In CVPR, pages 2064–2072 (2016)
- [2] Qi Li, Zhenan Sun, Ran He, and Tieniu Tan. "Deep supervised discrete hashing". In NIPS, pages 2482–2491 (2017)
- [3] Qing-Yuan Jiang and Wu-Jun Li. "Asymmetric deep supervised hashing". In AAAI (2018)
- [4] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. "Supervised discrete hashing". In CVPR, pages 37–45 (2015)
- [5] WU, Dayan, et al. "Deep incremental hashing network for efficient image retrieval". In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. p. 9069-9077 (2019)
- [6] VAN GANSBEKE, Wouter, et al. "Scan: Learning to classify images without labels". In: European conference on computer vision. Springer, Cham, p. 268-285 (2020)
- [7] YANG, Erkun, et al. "Semantic structure-based unsupervised deep hashing". In: Proceedings of the 27th international joint conference on artificial intelligence. p. 1064-1070 (2018)
- [8] REBUFFI, Sylvestre-Alvise, et al. "icarl: Incremental classifier and representation learning". In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. p. 2001-2010 (2017)