Self-Supervised Video Representation Learning with Meta-Contrastive Network

Yuanze Lin[1*] Xun Guo[2] Yan Lu[2]
[1]University of Washington, [2]Microsoft Research Asia
yuanze@uw.edu, {xunguo, yanlu}@microsoft.com

Abstract

*Self-supervised learning has been successfully applied to pre-train video representations, which aims at efficient adaptation from pre-training domain to downstream tasks. Existing approaches merely leverage contrastive loss to learn instance-level discrimination. However, lack of category information will lead to hard-positive problem that constrains the generalization ability of this kind of methods. We find that the multi-task process of meta learning can provide a solution to this problem. In this paper, we propose a **M**eta-**C**ontrastive **N**etwork (MCN), which combines the contrastive learning and meta learning, to enhance the learning ability of existing self-supervised approaches. Our method contains two training stages based on model-agnostic meta learning (MAML), each of which consists of a contrastive branch and a meta branch. Extensive evaluations demonstrate the effectiveness of our method. For two downstream tasks, i.e., video action recognition and video retrieval, MCN outperforms state-of-the-art approaches on UCF101 and HMDB51 datasets. To be more specific, with R(2+1)D backbone, MCN achieves Top-1 accuracies of **84.8%** and **54.5%** for video action recognition, as well as **52.5%** and **23.7%** for video retrieval.*

1. Introduction

Convolutional Neural Networks (CNNs) have brought unprecedented success for supervised video representation learning [4, 7, 8, 48, 29] . However, labeling large-scale video data requires huge human annotations, which is expensive and laborious. How to learn effective video representations by leveraging unlabeled videos is an important yet challenging problem. The recent progress of self-supervised learning for image provides an efficient solution to this problem [20, 43, 21, 5], which proposed to use contrastive loss [14, 15, 49, 51, 22] to discriminate different data samples.

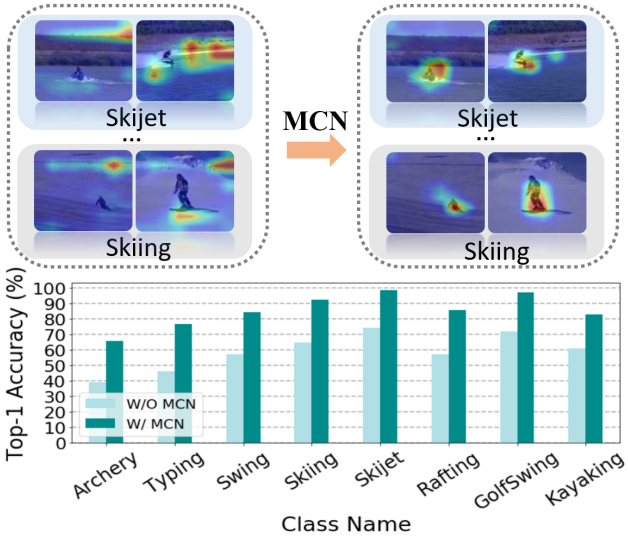

Figure 1. **Comparison between models trained without and with MCN on UCF101 [39].** The top row shows the activation maps produced by conv5 layer of R(2+1)D backbone using the method of [55]. By using our proposed MCN, the learned representations can capture motion areas more accurately. The bottom row shows top-1 accuracies of models trained without and with MCN approach.

This instance-based contrastive learning has also been applied to videos as pre-training, and achieved excellent performance on downstream tasks such as video action recognition and video retrieval [18, 42, 31]. However, it has the inherent limit of lacking the common category information. The instance-based discrimination process takes each video sample as an independent class, so that distance between two video samples will be pushed away by contrastive loss even if they belong to the same category. This drawback reduces the generalization of the pre-training parameters. Consequently, the efficiency of the supervised fine-tuning for the downstream tasks will also be damaged. How to improve the generalization of contrastive self-supervised learning and make the learned parameters easily adapt from pre-training domain to fine-tuning domain for various new tasks is still challenging.

Meta learning has demonstrated the capability of fast

*The work was done when the author was with MSRA as an intern.

adaptation on new tasks with only a few training samples. The characteristic of meta learning, specifically model-agnostic meta learning (MAML) [10], might help contrastive self-supervised video learning in two aspects. Firstly, instance-based discrimination takes each video as a class, so that it is convenient to create numerous sub-tasks for meta learning to improve the model generalization. Secondly, the goal of meta learning is "learn to learn", which means that it provides good initialization for fast adaptation on a new task. This perfectly meets the requirements of contrastive video representation learning, which is taken as a pre-training method. Therefore, combining meta learning and self-supervised learning might benefit to video representation learning.

In this paper, we propose a novel **M**eta-**C**ontrastive **N**etwork (MCN), which leverages meta-learning to improve the generalization and adaptation ability of contrastive self-supervised video learning on downstream tasks. The proposed MCN contains two branches, *i.e.*, contrastive branch and meta branch, which establishes a multi-task learning process to enhance the instance discrimination. Meanwhile, we design a two-stage training process based on MAML to improve the learning capability of MCN. Our method outperforms state-of-the-art methods and achieves significant performance boost.

The main contributions of this paper are summarized as follows.

1) We propose a novel **M**eta-**C**ontrastive **N**etwork (MCN), which can significantly improve the generalization of the video representations learned in self-supervised learning manner.

2) We fully investigate the benefits of combining meta learning with self-supervised video representation learning and conduct extensive experiments to make proposed approach better understood.

3) We evaluate our method on mainstream benchmarks for action recognition and retrieval tasks, which demonstrate that our proposed method can achieve state-of-the-art or comparable performance with other self-supervised learning approaches.

2. Related Work

2.1. Pretext Tasks

Early self-supervised learning approaches mainly focus on designing handcrafted pretext tasks for images, such as predicting the rotation of transformed images [12], image jigsaw [6], count of learned features [35], image colorization [56], relative positions [6] and so on.

After that, many self-supervised learning approaches about video data flourish. Due to the extra temporal dimension of video data, there are many pretext tasks specifically designed for temporal prediction, such as frame rate prediction [47], pace prediction [2], frame ordering prediction [53, 28, 9] and motion statistics prediction [46].

These pretext tasks make models achieve better discriminative ability, which is important for downstream tasks.

2.2. Contrastive Self-Supervised Learning

Contrastive self-supervised learning has been proved great potential in unlabeled data [20, 43, 21, 5, 32, 16]. Thanks to contrastive self-supervised learning approaches, model can be empowered to distinguish samples from different domains without labels.

There are some prior works in this area. He *et al.* [20] proposed a momentum dictionary to store and pop out learned features on the fly for images, so that the number of stored features can be extremely expanded. Chen *et al.* [5] proposed a simplified constrastive self-supervised image learning framework including only major components that benefit the learned representations. Tian *et al.* [43] presented a contrastive multi-view coding (CMC) approach for video representation learning, which uses different views of input videos to maximize the instance-level distinction. Our approach adopts CMC with two views, *i.e.*, RGB view and residual view, as baseline.

2.3. Meta Learning

Numerous research about meta learning has been presented for few-shot tasks. Finn *et al.* [10] proposed an important meta learning method called model-agnostic meta learning (MAML), which can be combined with any learning approaches trained with gradient descent. Some variants of MAML, *e.g.,* Reptile [33] and iMAML [38], can not only significantly save the training time, but also achieve comparable performance with MAML.

Recently, researchers start to focus on applying meta learning approaches to computer vision tasks, such as object tracking and face recognition [13, 45, 50]. In this paper, we utilize MAML to improve the performance of contrastive self-supervised video learning. Different from prior efforts, we try to enhance the adaptation between self-supervised pre-training domain and supervised fine-tuning domain, which is more challenging.

3. Meta-Contrastive Network

In this section, we describe our proposed meta-contrastive network (MCN) in details. In section 3.1, we introduce the framework of MCN and the two-stage training process. In section 3.2, we elaborate the contrastive branch of MCN. In section 3.3, the details of meta branch are clarified. In section 3.4, we introduce the losses and optimizations in MCN. Finally, In section 3.5, implementation details are explained.

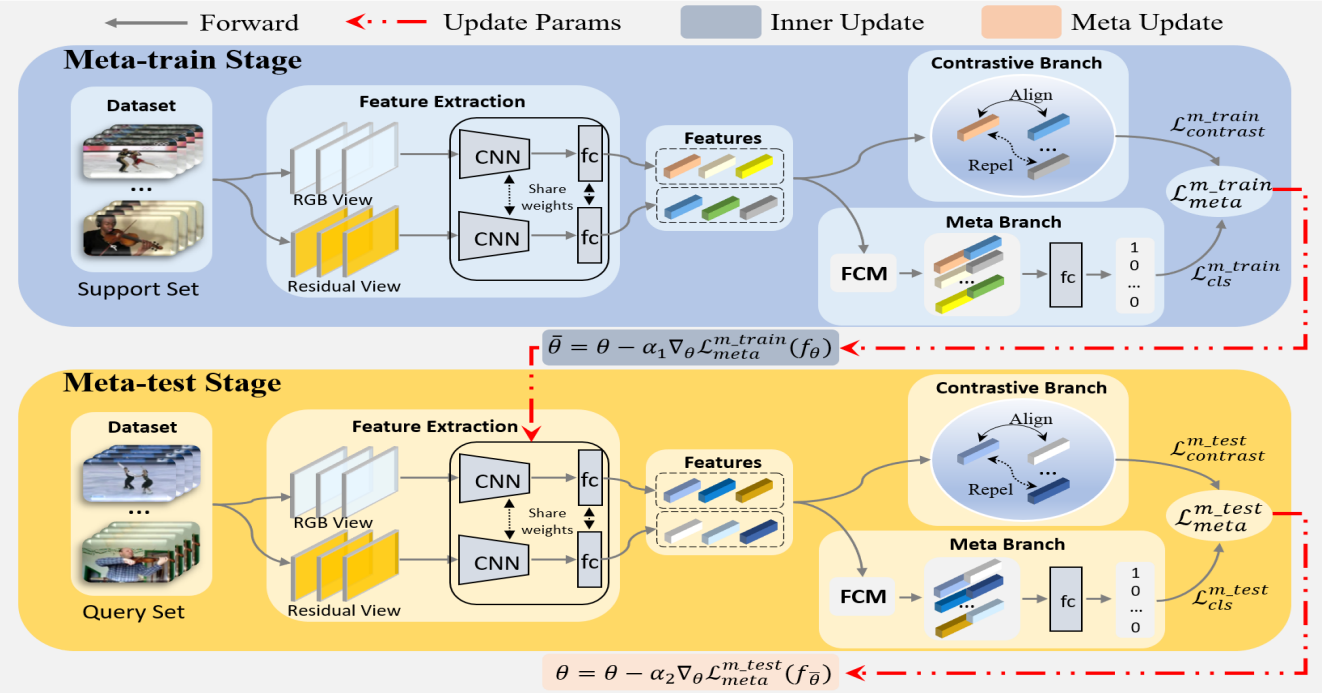

Figure 2. **The illustration of Meta-Contrastive Network**. For simplicity, only 3 input videos are used for illustration. There are two stages of MCN, including meta-train and meta-test stages. Model is parameterized by θ initially. α_1 and α_2 represent learning rates. Note that fully connected (fc) layer in meta branch is different from that in feature extraction module. FCM is feature combination module, which generates binary classification features for meta branch.

3.1. Framework

We build our framework by employing the contrastive multi-view coding (CMC) [43] as baseline. Multi-view input is proved to be efficient for instance-based video contrastive learning since different views, *i.e.*, transformations, from the same video can increase positive samples and make contrastive learning more efficient. We adopt two views in our framework, *i.e.*, RGB view and residual view, which have been proved to be extremely efficient views [41]. There are two branches in our framework as shown in Figure 2, *i.e.*, contrastive branch and meta branch. The contrastive branch performs contrastive learning, and the meta branch performs a couple of binary classifications for efficient meta learning. A binary classification is very similar with a pretext task that predicts whether the input two features come from the same video sample.

We employ a two-stage training process including meta-train and meta-test. Training data is split into train set, *i.e.*, *support set*, and test set, *i.e.*, *query set*. In meta-train stage, the videos from support set are used for inner update, in which the updated parameters will be used in meta-test stage for feature extraction. In meta-test stage, the videos from query set are used with the inner updated parameters for meta-update, which updates the initial parameters of meta-train stage for the next training iteration of MCN.

3.2. Contrastive Branch

The contrastive branch constructs a feature bank for positive and negative samples by collecting the extracted features for both of the two views, and calculates contrastive loss, *i.e.*, NCE loss [14]. The RGB view contains the sampled RGB video frames from a video clip, and the residual view contains the differences between two consecutive RGB frames. A residual frame is calculated as:

$$Frame_n^{Res} = |Frame_n^{RGB} - Frame_{n+1}^{RGB}|, \quad (1)$$

where $Frame_n^{Res}$ represents residual frame; $Frame_n^{RGB}$ represents RGB frame; n is the index of the sampled frame.

The reason why residual view is efficient may be that it can reflect the motions of the video clip to some extent and provides complementary information to RGB view. For example, when there are two different video clips with the same action, they may have similar residual view. This will implicitly increase the hard positive in contrastive learning process.

3.3. Meta Branch

Constrative learning suffers from hard-positive problem and hard-negative problem, which are even worse in self-supervised video learning. For example, there may exist videos that contain totally different scenes and objects but the same actions and events. There also exists videos that contain similar scenes and objects but different actions and events. Theoretically, meta learning can alleviate this problem due to the multi-task learning process. For this purpose, we design the meta branch consisting of a feature combination module (FCM) and several binary classification tasks, which can predict whether a feature pair belongs to the same video clip.

As shown in Figure 2, by concatenating two features of input video samples in FCM, several instance/binary classification tasks can be constructed in meta branch. The corresponding labels can be easily created for training. Figure 3 shows an example of creating classification task with FCM for two video samples v_1 and v_2. If one concatenated feature is from the different views of the same video, the label will be true, otherwise the label will be false.

We design the binary classification lies in two reasons. Firstly, the binary classification loss can be complementary with contrastive loss to better learn the instance discrimination. Secondly, the binary classification enables efficient combination of contrastive learning branch and meta learning branch to improve the generalization through multi-task learning process.

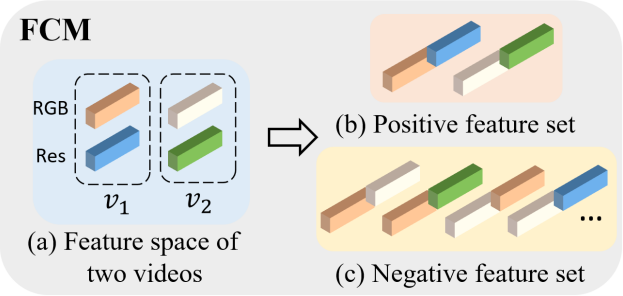

Figure 3. **An example of FCM**. v_1 and v_2 are two video samples. (a) is feature space of the two videos. RGB and Res mean features extracted from RGB view and Residual view respectively. (b) is positive feature set, whose label is true. (c) is negative feature set, whose label is false.

3.4. Meta Loss and Optimization

In order to ease the training process of MCN method, we incorporate both metric losses, *i.e.*, contrastive loss and cross-entropy loss, *i.e.*, classification loss, and propose the combined meta loss for final optimization.

Contrastive Loss. Contrastive learning aims to separate features from different samples. In our method, we employ the contrastive loss form from CMC [43] as the objective of the contrastive branch. Specifically, two different views of the same sample, *e.g.*, $\{x_i^1, x_i^2\}$, are treated as positive, while the views from different samples, *e.g.*, $\{x_i^1, x_j^2\}$ $(i \neq j)$, are regarded as negative.

A value function h_θ is used so that positive pairs have high score, and negative pairs obtain low score. To be more concrete, after feature z_i^1 is extracted by the model, function h_θ is trained on a feature set $Z = \{z_1^2, z_i^2, ..., z_{k+1}^2\}$, which consists of one positive sample z_i^2 and k negative samples, so that the positive sample can be easily picked out from Z. The contrastive loss can be formulated as:

$$L_{contrast} = -log\frac{h_\theta(\{z_i^1, z_i^2\})}{\sum_{j=1}^{k+1} h_\theta(\{z_i^1, z_j^2\})}, \tag{2}$$

where $L_{contrast}$ denotes contrast loss for contrastive branch, and k is the number of negative samples. z_i^1 and z_i^2 mean extracted features of two different views from ith sample. $h_\theta(\cdot)$ can be formulated as:

$$h_\theta(\{z_i^1, z_j^2\}) = exp(\frac{z_i^1 \cdot z_j^2}{\|z_i^1\| \cdot \|z_j^2\| \cdot \tau}), \tag{3}$$

where $h_\theta(\cdot)$ is cosine similarity of two features and τ is the parameter for dynamically controlling the range.

Classification Loss. Meta branch of MCN performs instance/binary classification. We use binary cross-entropy loss (BCE) as our classification loss, which can be formulated as:

$$L_{cls} = \sum_{i=1}^{N} -y_i log\hat{y} - (1 - y_i)log(1 - \hat{y}_i), \tag{4}$$

where N is the number of concatenated features. In our approach, FCN concatenates features from 4 video clips in each batch and each video has two features. y_i is the label of ith concatenated feature. $\hat{y}_i$ is the output of the fully connected layer of meta branch.

Meta Loss. Contrastive loss and classification loss from the two branches are combined together to get the final meta loss, which is defined as:

$$L_{meta} = \alpha \cdot L_{cls} + (1 - \alpha) \cdot L_{contrast}, \tag{5}$$

where α is a hyper-parameter used to control the relative impact of binary classification loss L_{cls} and contrast loss $L_{contrast}$ respectively. The meta loss is used to update weights in meta-train and meta-test stages.

Optimization. During meta-train or meta-test stage, meta loss L_{meta} is used to optimize model parameters θ with gradient descent. In meta-train stage, L_{meta} achieved from support set is used to get updated parameters $\bar{\theta}$, which is denoted as inner update. And in meta-test stage, L_{meta}

obtained from query set is used to update θ, which is denoted as meta update. Step sizes of gradient descent for the two optimization stages are the same as learning rate.

3.5. Implementation Details

We will further explain implementation details of MCN in this section. In specific, our proposed MCN consists of two stages, *i.e.*, meta-train and meta-test stages. The whole process of MCN is explained as follows.

Initialize. A pre-trained model $f(\theta)$ parametrized by θ, dataset D, support set D_s, query set D_q, batch size B, $D = D_s \cup D_q$.

Input. Sample B input videos X_{sup} from support set D_s. Sample B input videos X_{que} from query set D_q.

Meta-train. Feed sampled videos X_{sup} into network $f(\theta)$ to extract features. Use these features to compute contrastive loss $L_{contrast}^{m\_train}$ by Equation 2. Use FCM to combine these features and get a new feature set S_{m_train}. Set S_{m_train} is used to compute classification loss $L_{cls}^{m\_train}$ by Equation 4, then $L_{cls}^{m\_train}$ and $L_{contrast}^{m\_train}$ are used to compute $L_{meta}^{m\_train}$ by Equation 5.

Inner Update. Use $L_{meta}^{m\_train}$ to update model parameters by gradient descent. The updated process is $\bar{\theta} = \theta - \alpha_1 \nabla_\theta L_{meta}^{m\_train}(f_\theta)$, where α_1 is the same as learning rate.

Meta-test. Feed sampled videos X_{que} into network $f(\bar{\theta})$ to extract features. Use these features to compute contrastive loss $L_{contrast}^{m\_test}$ by Equation 2. Use FCM to combine these features and get a new feature set S_{m_test}. Set S_{m_test} is used to compute classification loss $L_{cls}^{m\_test}$ by Equation 4, then $L_{cls}^{m\_test}$ and $L_{contrast}^{m\_test}$ are used to compute $L_{meta}^{m\_test}$ by Equation 5.

Meta Update. Use $L_{meta}^{m\_test}$ to update the final model parameters by gradient descent. Corresponding updated process is $\theta = \theta - \alpha_2 \nabla_\theta L_{meta}^{m\_test}(f_{\bar{\theta}})$, where α_2 is the same as learning rate.

4. Experiments

4.1. Datasets

We evaluate our approach on three video classification datasets including UCF101 [39], HMDB51 [27] and Kinetics-400 [24].

UCF101 [39] is a dataset that has 101 action categories, containing 13320 videos totally. There are 3 splits on this data set [2, 53]. In our experiments, we use train split 1 as self-supervised pre-training dataset, and train/test split 1 for fine-tuning/evaluation.

HMDB51 [27] has around 7000 videos with 51 video action classes, which is relatively small compared to UCF101 and Kinetics [24]. It also has 3 splits. We use split 1 for fine-tuning and evaluation.

Kinetics-400 [24] is a popular benchmark for action recognition collected from Youtube, which contains 400 action categories. There are totally 300K video samples, which are divided to 240K, 20K and 40K for training, validation and test sets respectively. In our paper, we only use train split as our pre-training dataset

4.2. Experimental Setup

Data Pre-processing. We randomly sample 32 continuous frames from each video as the input of MCN. If the original videos are not long enough, the first frame will be repeated. The sampled original frames are treated as RGB view, and the residual frames generated by Equation 1 is treated as residual view. The original frames will be randomly cropped and resized into 128×128. Meanwhile, Gaussian blur, horizontal flips and color jittering are also used for augmentation.

Backbones. Three main-stream network structures, *i.e.*, S3D [52], R3D-18 [19, 44] and R(2+1)D [44] are used as the backbones of MCN in ablation experiments. For video action recognition and video retrieval tasks, only the results of R3D-18 and R(2+1)D are reported.

Self-Supervised Learning. We train our models using 4 NVIDIA Tesla P40 around 500 epochs. Initial learning rate is 0.01 and weight decay is 0.001. α is set to 0.2. We use the batch sizes of 28 and 80 for R(2+1)D and R3D-18 respectively.

Fine-tuning. After finishing self-supervised learning stage, we fine-tune the pre-trained models on UCF101 or HMDB51 around 300 epochs. A new fully connected layer will be added to the end of the pre-trained backbone for classification. Learning rate is set as 0.02. And the batch sizes are 72 and 200 for R(2+1)D and R3D-18 respectively.

Evaluations. Evaluations of proposed method are conducted on video action recognition and video retrieval tasks. For video action recognition, the top-1 accuracies on UCF101 [39] and HMDB51 [27] are reported. In order to further validate our proposed MCN, we also show the results of linear probe results, in which the weights of self-supervised learning model are fixed and only the fully-connected layers for supervised classification are fine-tuned. For video retrieval, top-1, top-5, top-10, top-20 and top-50 accuracies are compared with existing approachs.

4.3. Ablation Studies

To fully investigate and understand the concept of MCN, we conduct ablation experiments to demonstrate how each design of MCN affects the overall performance.

Comparison with Baseline. We compare the action recognition results of self-supervised training with and without MCN in Table 1. We use three backbones, *i.e.*, S3D, R(2+1)D and R3D-18, to demonstrate the performance boost of MCN. As shown in this table, the accura-

cies of baseline with S3D, R(2+1)D and R3D-18 are 76.7%, 77.3% and 78.6% on UCF101 dataset respectively. while the accuracies of MCN are **82.9%**, **84.8%** and **85.4%** respectively. Relevant results of HMDB51 dataset can also be observed. There is consistent performance boost when using MCN on different backbones and data sets.

Furthermore, we also evaluate the linear probe results in Table 2, in which only the fully connected layers are fine-tuned. Significant performance boost of MCN can also be observed on both UCF101 and HMDB51 dataset.

Methods	Backbone	UCF101(%)	HMDB51(%)
Ours (Baseline)	S3D	76.7	45.5
Ours (+ MCN)	S3D	**82.9**	**53.8**
Ours (Baseline)	R(2+1)D	77.3	46.2
Ours (+ MCN)	R(2+1)D	**84.8**	**54.5**
Ours (Baseline)	R3D-18	78.6	47.1
Ours (+ MCN)	R3D-18	**85.4**	**54.8**

Table 1. Comparisons between MCN and baseline with different backbones on video action recognition task.

Methods	Backbone	UCF101(%)	HMDB51(%)
Ours (Baseline)	S3D	62.4	33.5
Ours (+MCN)	S3D	**71.6**	**40.8**
Ours (Baseline)	R(2+1)D	64.2	35.6
Ours (+MCN)	R(2+1)D	**72.4**	**41.2**
Ours (Baseline)	R3D-18	64.6	37.3
Ours (+MCN)	R3D-18	**73.1**	**42.9**

Table 2. Linear probe evaluation results of different backbones on video action recognition task.

Influence of α. As depicted in Equation 5, α is introduced to modulate meta loss. We also conducted experiments to demonstrate the influence of this hyper-parameter. Table 3 shows the results of 4 settings of α with R(2+1)D backbone.

Settings	UCF101(%)
$\alpha = 0.1$	84.1
$\alpha = 0.2$	**84.8**
$\alpha = 0.3$	83.4
$\alpha = 0.4$	82.7

Table 3. Results of different α settings on UCF101 dataset.

We can observe that setting α as 0.2 shows the best performance. Therefore, we set α as 0.2 in all our experiments.

Influence of Input Frames. For self-supervised video representation learning, the number of input frames for each video clip may affect the final performance. Therefore, we tested different numbers for quantitative analysis. We firstly pre-train models on UCF101 dataset, then fine-tune the models for video action recognition task.

In Table 4, we can see that more input frames bring better performance. As the input length increases, MCN takes additional improvement.

Methods	Input Frames	UCF101(%)
Baseline	16	74.6
MCN	16	**81.3**
Baseline	32	77.3
MCN	32	**84.8**
Baseline	64	80.6
MCN	64	**86.7**

Table 4. Results of different input frames for MCN and baseline on video action recognition task.

Influence of Individual Component. We also test each component of MCN to figure out their contributions to the final performance. The results of video action recognition on UCF101 are demonstrated in Table 5. R(2+1)D is selected as backbone.

As shown in Table 5. CL represents contrastive loss. BL represents binary loss from proposed meta branch. Combining CL and BL without meta stages takes **1.9%** accuracy improvement. By adding meta stages, additional **5.6%** improvement is achieved, which proves the efficiency of meta learning. These experiments can demonstrate the effectiveness of proposed MCN method.

CL	BL	Meta Stages	UCF101(%)
✓			77.3
✓	✓		79.2
✓	✓	✓	**84.8**

Table 5. Ablation study for different components of MCN on video recognition task.

4.4. Evaluation of MCN

In this section, we compare the performance of our proposed method with other state-of-the-art approaches. We show the evaluations on two downstream tasks including video action recognition and video retrieval. R3D-18 and R(2+1)D are used as backbones for the comparisons.

Video Action Recognition. Considering that we only use RGB information in our experiments, we didn't include the approaches with multi-modality [26, 1, 36, 31]. CoCLR also [18] demonstrates excellent performance by co-training RGB and optical flow samples. In this paper, we only include the RGB-only results of CoCLR for fair comparison.

We first compare our linear probe evaluation results with other state-of-the-art approaches so that we can verify the transferability of the video representations learned with our approach. Results in Table 6 demonstrate that the proposed MCN method outperforms state-of-the-art approaches on both UCF101 and HMDB51.

Methods	UCF101(%)	HMDB51(%)
CBT [40]	54.0	29.5
MemDPC [17]	54.1	30.5
CoCLR [18]	70.2	39.1
Ours (R(2+1)D)	**72.4**	**42.2**
Ours (R3D-18)	**73.1**	**42.9**

Table 6. Linear probe comparisons with state-of-the-art methods on UCF101 and HMDB51 datasets.

Methods	Backbone	Resolution	UCF101	HMDB51
Jigsaw[34]	UCF101	225	51.5	22.5
OPN [28]	VGG	227	56.3	22.1
Mars [46]	C3D	112	58.8	32.6
CMC [43]	CaffeNet	128	59.1	26.7
ST-puzzle [25]	R3D	224	65.0	31.3
VCP [30]	R(2+1)D	112	66.3	32.2
VCOP [53]	R(2+1)D	112	72.4	30.9
PRP [54]	R(2+1)D	112	72.1	35.0
IIC [42]	R3D	112	74.4	38.3
PP [47]	R(2+1)D	112	75.9	35.9
CoCLR [18]	S3D	128	81.4	52.1
Ours	R(2+1)D	128	**84.8**	**54.5**
Ours	R3D	128	**85.4**	**54.8**

Table 7. Comparisons with state-of-the-art methods for video action recognition on UCF101 and HMDB51 datasets (models are pre-trained on UCF101).

Methods	Backbone	Resolution	UCF101	HMDB51
3D-RotNet [23]	R3D	112	62.9	33.7
ST-Puzzle[25]	R3D	224	63.9	33.7
DPC [16]	R2D-3D	128	75.7	35.7
SpeedNet [2]	S3D-G	224	81.1	48.8
PP [47]	R(2+1)D	112	75.9	35.9
CoCLR [18]	S3D	128	87.9	54.6
CVRL [37]	R3D	224	92.1	65.4
Ours	R(2+1)D	128	**89.2**	**58.8**
Ours	R3D	128	**89.7**	**59.3**

Table 8. Comparisons with state-of-the-art methods for video action recognition on UCF101 and HMDB51 datasets (models are pre-trained on Kinetics-400).

We then compare the results of fine-tuning all parameters with other state-of-the-art methods with different pre-training datasets. In specific, we pre-train our models on both UCF101 and Kinetics-400, and then fine-tune the pre-trained models on UCF101 and HMDB51. Table 7 and Table 8 show the results respectively. From the tables, we can observe that the results pre-trained on Kinetics-400 are much better than that pre-trained on UCF101. Kinetics contains much more videos than UCF101. The results demonstrates that MCN can better leverage large volume of unlabeled videos. In both tables, our method outperforms or achieves comparable performance with other state-of-the-art self-supervised approaches. In Table 8, CVRL shows better result than ours. This may be due to three reasons: (1) larger input image resolution (224×224) compared with ours (128×128); (2) more powerful and deeper backbone network (R3D-50) than ours (R(2+1)D and R3D-18); (3) more efficient data augmentation approaches. These experimental results can shed a light for combining meta learning with self-supervised learning approaches.

Methods	top1	top5	top10	top20	top50
Jigsaw [34]	19.7	28.5	33.5	40.0	49.4
OPN[28]	19.9	28.7	34.0	40.6	51.6
Büchler [3]	25.7	36.2	42.2	49.2	59.5
VCOP [53]	10.7	25.9	35.4	47.3	63.9
VCP [30]	19.9	33.7	42.0	50.5	64.4
CMC [43]	26.4	37.7	45.1	53.2	66.3
PP [47]	31.9	49.7	59.2	68.9	80.2
IIC [42]	42.4	60.9	69.2	77.1	86.5
CoCLR [18]	53.3	69.4	76.6	82.0	-
Ours (R(2+1)D)	52.5	**69.5**	**77.9**	**83.1**	**89.3**
Ours (R3D)	**53.8**	**70.2**	**78.3**	**83.4**	**89.7**

Table 9. Comparisons with state-of-the-art approaches for video retrieval on UCF101 dataset.

Methods	top1	top5	top10	top20	top50
VCOP [53]	7.6	22.9	34.4	48.8	68.9
VCP [30]	7.6	24.4	36.3	53.6	76.4
CMC [43]	10.2	25.3	36.6	51.6	74.3
PP [47]	12.5	32.2	45.4	61.0	80.7
IIC [42]	19.7	42.9	57.1	70.6	85.9
CoCLR [18]	23.2	43.2	53.5	65.5	-
Ours (R(2+1)D)	**23.7**	**46.5**	**58.9**	**72.4**	**87.3**
Ours (R3D)	**24.1**	**46.8**	**59.7**	**74.2**	**87.6**

Table 10. Comparisons with state-of-the-art approaches for video retrieval on HMDB51 dataset.

Video Retrieval. In addition to video action recognition task, we also evaluate the performance of MCN on video retrieval task, which can better reflect the semantic-level learning capability. Instead of using RGB and residual views, RGB and flow views of original video clips are considered for video retrieval, in which selected flow view is the vertical dimension of optical flow. We extract optical flow of input videos by using un-supervised TV-L1 algorithm [11]. Video retrieval task is conducted with extracted features from pre-trained models without extra fine-tuning stages. We take every video from test set to query k nearest videos from the training set based on its extracted features. When the class of retrieval video is the same as that of the qeury video, this retrieval result is considered correct. The top-1, top-5, top-10, top-20, and top-50 retrieval accuracies have been shown in our experiments.

As shown in Table 9, when compared with other state-of-art methods, our method achieves superior or comparable performance in UCF101 dataset. We observe that the top-1 accuracy of CoCLR is slightly better than our R(2+1)D backbone. Actually, our method is orthogonal to CoCLR.

Figure 4. **Video retrieval examples on UCF101.** The first column represents query videos from the test split, and the remaining columns are top-3 results retrieved by the models trained without and with MCN from the training split. The class name of each video is shown in bottom. Red fonts denote wrong video retrieval results.

In other words, MCN can take the model trained by CoCLR as baseline to take additional improvement. The results of HMDB51 dataset have been shown in Table 10, which demonstrate the superior performance of our proposed MCN.

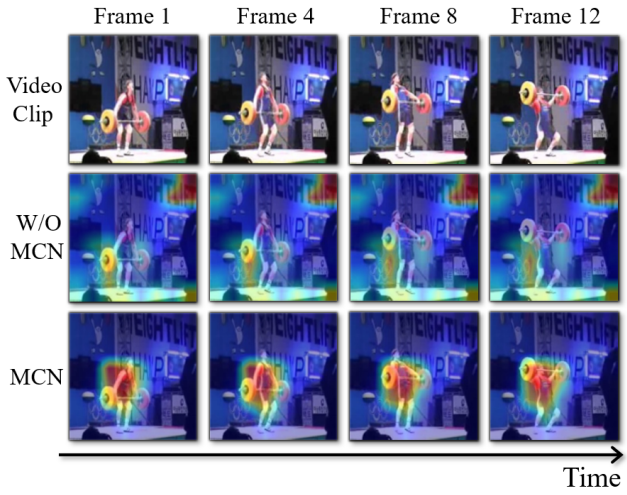

Figure 5. **Activation maps produced from conv5 layer of R(2+1)D backbone.** The maps are generated with 32-frames input and the method in [55] is used. $1st$, $4th$, $8th$ and $12th$ frames of the video clip are illustrated. The three rows represent original video clip, activation maps produced by models trained without and with MCN respectively.

In Figure 4, retrieval results of models with and without MCN are visualized. R(2+1)D is used as backbone. Video clips from UCF101 test set are used to query 3 nearest videos from UCF101 training set. We can clearly observe that the learned representations with MCN can query videos with the same classes more accurately.

4.5. Visualization

In this section, we visualize activation maps of MCN in Figure 5, so that we can intuitively understand what has been improved during self-supervised learning process. We use the method in [55] to visualize activation maps from conv5 layer of pre-trained R(2+1)D backbone.

It is interesting to observe that, the model trained without MCN may focus on the irrelevant areas, while MCN can accurately pay attention to the motion areas of video clips. This is essential for action recognition. For example, in the first row of Figure 5, we can clearly see that a person is doing a clean and jerk. The learned representations by MCN can focus more on his action areas, such as hands and shoulders.

5. Conclusion

In this paper, we propose a novel **M**eta-**C**ontrastive **N**etwork (MCN), which leverages meta-learning to improve the generalization and adaptation ability of contrastive self-supervised video learning on downstream tasks. The proposed MCN contains two branches, *i.e.*, contrastive branch and meta branch, which combine NCE loss and binary classification loss together to enhance the instance discrimination. Meanwhile, we design a two-stage training process based on MAML to improve the learning capability of MCN. Our method outperforms state-of-the-art methods and achieves significant performance boost. To our best knowledge, this is the first time that contrastive self-supervised video learning is combined with meta learning. We also hope our work can inspire more researchers who have interests on this field.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*, 2019. 6

[2] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9922–9931, 2020. 2, 5, 7

[3] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 770–786, 2018. 7

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2

[6] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015. 2

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 1

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 1

[9] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3636–3645, 2017. 2

[10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017. 2

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 7

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[13] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6172, 2020. 2

[14] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 1, 3

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 1

[16] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2, 7

[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv preprint arXiv:2008.01065*, 2020. 7

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Neurips*, 2020. 1, 6, 7

[19] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 5

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[21] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR, 2020. 1, 2

[22] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 1

[23] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018. 7

[24] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 5

[25] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019. 7

[26] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018. 6

[27] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011. 5

[28] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 667–676, 2017. 2, 7

[29] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 1

[30] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11701–11708, 2020. 7

[31] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 1, 6

[32] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 2

[33] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018. 2

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. 7

[35] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017. 2

[36] Mandela Patrick, Yuki M Asano, Ruth Fong, Joao F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. *arXiv preprint arXiv:2003.04298*, 2020. 6

[37] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020. 7

[38] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. *arXiv preprint arXiv:1909.04630*, 2019. 2

[39] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1, 5

[40] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 7

[41] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Rethinking motion representation: Residual frames with 3d convnets for better action recognition. *arXiv preprint arXiv:2001.05661*, 2020. 3

[42] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2193–2201, 2020. 1, 7

[43] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 1, 2, 3, 4, 7

[44] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 5

[45] Guangting Wang, Chong Luo, Xiaoyan Sun, Zhiwei Xiong, and Wenjun Zeng. Tracking by instance detection: A meta-learning approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6288–6297, 2020. 2

[46] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4006–4015, 2019. 2, 7

[47] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision*, pages 504–521. Springer, 2020. 2, 7

[48] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 1

[49] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2015. 1

[50] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9925–9934, 2019. 2

[51] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 1

[52] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018. 5

[53] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2, 5, 7

[54] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised

spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6548–6557, 2020. 7

[55] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 1, 8

[56] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2