Computer Science and Application 计算机科学与应用, 2021, 11(3), 604-617
Published Online March 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.113062

Hans汉斯

融合GRU和非极大值抑制的视频摘要生成模型

陈周元，陈平华，申建芳

广东工业大学计算机学院，广东 广州
Email: 1623240122@qq.com

收稿日期：2021年2月20日；录用日期：2021年3月15日；发布日期：2021年3月24日

摘　要

现有视频摘要生成模型存在计算量大，冗余帧带来的性能损耗大，模型效果不稳定等问题。基于此，提出融合GRU和非极大值抑制的视频摘要生成模型。所提模型对视频帧之间的特征关系进行建模，在获取帧级重要性得分模块中，提出一种融入GRU和注意力机制的Seq2Seq模型，增强帧与帧之间的时域特征关系影响，并且有效减少模型计算量，提高模型在反向传播时的收敛速度；在获取视频摘要模块中，提出基于非极大值抑制的关键序列生成算法，有效去除冗余帧。通过在多个数据集上与现今主流的视频摘要生成模型比对，显示所提模型在F-score和KFRR两个评估指标上均有不同程度的提升，表明其所生成的视频摘要具有更强的内容概括能力，并且模型在各种数据状况下具有较高的稳定性。

关键词

视频摘要，卷积神经网络，GRU网络，注意力机制，非极大值抑制

Model of Video Summarization Integrating GRU and Non-Maximum Suppression

Zhouyuan Chen, Pinghua Chen, Jianfang Shen

School of Computer, Guangdong University of Technology, Guangzhou Guangdong
Email: 1623240122@qq.com

Received: Feb. 20th, 2021; accepted: Mar. 15th, 2021; published: Mar. 24th, 2021

Abstract

Existing models of video summarization have problems such as too much calculation, large negative impact caused by redundant frames and unstable model effects. To deal with these problems, model of video summarization integrating GRU and non-maximum suppression is proposed. In the module of getting frame-level importance score, this paper proposes a kind of Seq2Seq model in-

文章引用: 陈周元, 陈平华, 申建芳. 融合 GRU 和非极大值抑制的视频摘要生成模型[J]. 计算机科学与应用, 2021, 11(3): 604-617. DOI: 10.12677/csa.2021.113062

corporating GRU and attention mechanism, which enhances the influence of the time-domain feature relationship between frames and effectively reduces the amount of model calculations, improving convergence speed during back propagation. In the module of summarizing video, this paper proposes a key sequence generation algorithm based on non-maximum suppression, which effectively removes redundant frames. By comparing with the current mainstream models of video summarization on multiple datasets, it is shown that the proposed model has different degrees of improvement in the two evaluation indicators of F-score and KFRR, indicating that the generated video summarization has stronger content generalization ability, and the model has high stability under various data conditions.

Keywords

Video Summarization, Convolutional Neural Networks, GRU Network, Attention Mechanism, Non-Maximum Suppression

1. 概述

近年来，大量的数字视频被生产并应用于教育、娱乐、监视、信息存档等领域，数字视频已经成为人们视觉信息的最重要来源之一。大量的视频数据增加了人们对于数字视频浏览、筛选和存储的压力。依照传统的方法，用户只能通过视频标题、简介、标签等有限的外部资源信息间接了解视频内容，而对视频本身的内容没有一个直观的理解。视频摘要生成技术在提高用户对视频内容信息的获取上发挥了重要作用，该技术通过分析一定长度的视频数据中信息的稀疏性，从原始视频数据中选取具有代表性的、有意义的部分，将它们以某种方式组合并生成紧凑的、用户可读的缩略数据，使用户在更短的时间内快速理解视频[1]。

根据生成摘要过程是否需要标注数据，生成视频摘要的研究可以分为无监督学习的生成方法和有监督学习的生成方法。无监督学习生成方法通过自定义直观的标准来挑选关键帧或关键镜头，并组合成相应的视频摘要；有监督学习生成方法从人工创建的摘要中学习特征信息，拟合人类对输入视频进行总结的方式。在实际应用中，这些模型表现出不错的效果，但是它们仍然存在一些不可忽视的问题：一些研究者为了提高模型的性能，不断增加模型的复杂度，使得模型的计算量大大增加，在训练数据不足的情况下难以达到理想的收敛效果；并且大多数模型对冗余帧的判定和处理的效果并不理想。

针对上述存在的问题，本文提出融合 GRU 和非极大值抑制的视频摘要生成模型。所提模型将视频数据看成以帧为单位的序列数据，首先用 GoogLeNet 网络提取每一帧的图片特征，其次将帧特征序列传入包含 GRU 和注意力机制的序列到序列(Sequence-to-Sequence, Seq2Seq)模型中，提取帧级重要性得分，最后通过基于非极大值抑制(Non-Maximum Suppression, NMS)的算法去除冗余帧并生成关键帧序列和关键镜头序列。

本文的研究主要有如下三点贡献：其一，提出一种融入 GRU 和注意力机制的 Seq2Seq 模型，使模型在处理视频帧序列时能最大程度保留帧与帧之间的长距离影响因素，同时减少模型的参数，提高反向传播时的收敛速度；其二，提出基于非极大值抑制的算法，有效地处理冗余帧，获得更具代表性的视频摘要；其三，在 SumMe、TVSum 和 VSUMM 三种数据集中验证了模型的有效性。

本文后面的组织架构为：第 1 节介绍与视频摘要相关的研究工作，第 2 节介绍所提的融合 GRU 和非极大值抑制的视频摘要生成模型及算法原理，第 3 节进行实验对比和分析，第 4 节进行工作总结与展望。

2. 相关工作

早期的视频摘要方法大多基于无监督学习方法[2] [3] [4] [5] [6]，通过外观或运动特征等底层视觉信息来提取视频摘要，其中聚类算法的使用最为广泛。对于基于无监督聚类的方法，基本思想是通过将相似的帧或镜头聚集在一起，然后在每个聚类中选取特定数量的帧或镜头进行组合生成摘要。该方法关键点在于模型需要选择可以判断帧相似的特征(例如，颜色分布，亮度，运动矢量)，进而建立可用于测量相似性的不同标准。聚类算法生成摘要通常需要花费较长时间，文献[6]中提到，通过聚类算法生成摘要所花费的时间大约是原始视频时长的数倍。除此之外，由于聚类算法通常只关注视频帧的重要性程度，容易忽略掉视频的时域信息对摘要生成的影响。

随着深度神经网络的不断发展，有监督学习的视频摘要方法开始得到关注。有监督的视频摘要方法能准确地捕获视频帧的选择标准，并输出与人类对视频内容的语义理解更加一致的帧子集。在有监督学习方向上，文献[7] [8] [9] [10]认为同类别的视频数据具有相似的上下文结构，故利用模型从相关的照片或从属于特定事件类别的视频中学习特征，最终得到对特定领域更有表征性的摘要；文献[11] [12]利用原始视频和编辑过的视频进行相对排名，省去了标记数据的繁琐工作。

近年来，基于 Seq2Seq 模型的视频摘要生成模型[13]-[18]受到研究者的广泛关注。文献[14]使用融入长短期记忆(Long Short-Term Memory, LSTM)单元的 Seq2Seq 模型自动生成帧级重要性得分，并通过交叉熵和行列式点过程(Determinantal Point Process, DPP)筛选关键帧。文献[15]在上述 Seq2Seq 模型中引入视觉注意力机制，提高历史解码信息的利用率，并使用背包算法筛选关键帧。与上述方法不同，本文在 Seq2Seq 模型中使用 GRU 代替 LSTM 单元，提高训练效率，使模型能达到更好的收敛效果，同时使用非极大值抑制筛选关键帧，达到更好的去冗余效果。

3. 融合 GRU 和非极大值抑制的视频摘要生成模型

本文所提模型包括特征提取模块、重要性得分生成模块和摘要生成模块，其整体结构如图 1 所示。特征提取模块负责对输入的原始视频下采样和特征提取，将获得的视频帧序列输出到下一模块；重要性得分生成模块负责分析并捕捉视频帧序列的上下文信息，生成帧级重要性得分；摘要生成模块根据重要性得分结果，使用非极大值抑制去除冗余帧，生成相应的关键帧序列和关键镜头序列。

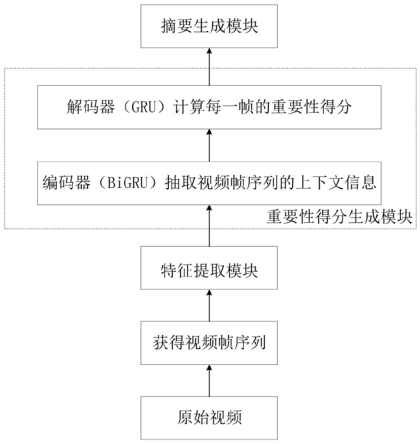

Figure 1. Model structure diagram of video summarization integrating GRU and non-maximum suppression
图 1. 融合 GRU 和非极大值抑制的视频摘要生成模型结构图

3.1. 特征提取模块

本文将 GoogLeNet [19]作为帧特征提取器，负责提取视频数据每一帧的图片特征。与其他卷积神经网络不同，GoogLeNet 在网络层之间添加了多个 inception 块，一定程度上减少网络层数，在相同的计算量下具有更好的分类性能。GoogLeNet 中的 inception 结构如图 2 所示，通过引入 Inception 块，GoogLeNet 在相同尺寸的感受野中能叠加更多的卷积，提取到更丰富的帧特征；同时 Inception 中的多个 1 × 1 卷积层能有效降低模型的维度，防止训练阶段出现过拟合。

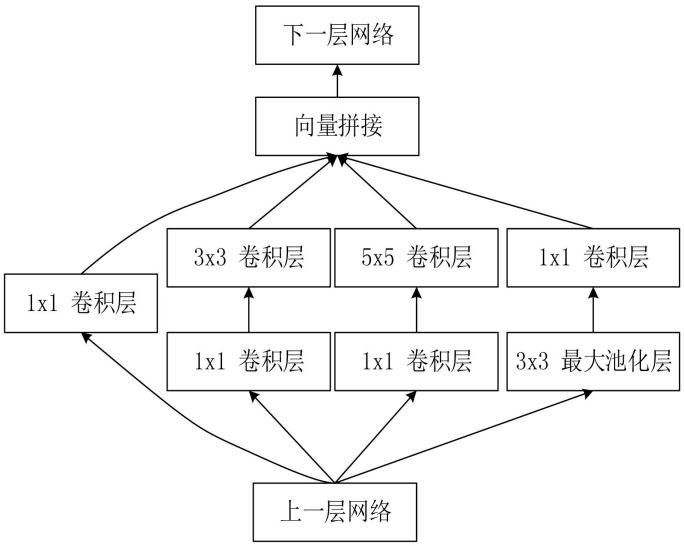

Figure 2. Structure diagram of inception
图 2. Inception 结构图

由于所用数据集中的部分视频数据包含较多冗余帧，相近的视频帧具有相近的特征，故本文首先需要对原始视频帧序列进行下采样。根据时间顺序生成原始视频的视频帧序列，记为 V，则下采样后的视频帧序列 V' 由 V 每 30 帧中随机抽取 2 帧组成。然后将 V' 输入到经过预训练的 GoogLeNet 中，获得帧级图片特征序列，记为 $Z=\{F_0,F_1,F_2,\cdots,F_{n-1}\}$，其中 F_i 表示下采样后第 i 帧的特征向量，n 表示下采样后的总帧数。

3.2. 基于 GRU 的 Seq2Seq 模型计算重要性得分

本文将获取帧级重要性得分任务看成机器翻译任务，将获取帧级重要性得分过程按照机器翻译过程进行处理。获取帧级重要性得分模块的输入是视频帧特征序列 $Z=\{F_0,F_1,F_2,\cdots,F_{n-1}\}$，其中 F_i 表示第 i 帧的图片特征向量。Z 由上一模块的 GoogleNet 网络提取获得。获取帧级重要性得分模块的输出是重要性得分序列 $P=\{S_0,S_1,S_2,\cdots,S_{n-1}\}$，其中 S_i 表示第 i 帧的重要性得分。

即便 Seq2Seq 模型善于处理序列数据，它仍然存在一些弊端。Seq2Seq 模型的编码器将输入编码为固定大小状态向量的过程是一个信息有损压缩的过程，信息量越大，该转化向量的过程对信息的损失就越大；同时 Seq2Seq 模型中的循环神经网络(Recurrent Neural Network, RNN)在处理过长的序列时，若对当前状态有用的信息距当前状态的时间间隔较大，这些信息记录将变得模糊，导致在训练时出现梯度弥散问题且计算效率低下；除此之外，模型连接编码器和解码器的模块组件只有一个固定大小的状态向量，导致解码器无法直接去关注到输入信息的更多细节。

针对以上提到的问题，本文在下面分别对编码器和解码器做出相应的改进。

3.2.1. 编码器改进：双向 GRU 网络提取上下文信息

本文采用双向 GRU 网络提取视频帧的上下文信息。GRU (Gate Recurrent Unit) [20]作为一种循环神经网络的变体结构，通过控制重置门(Reset Gate)和更新门(Update Gate)来处理当前节点的上一状态信息(即上一视频帧的有效信息)和上一层输入数据(即当前视频帧的图片特征)，并获得当前节点的状态信息(即当前视频帧的有效信息)。GRU 网络可以解决 RNN 中存在的长期记忆模糊问题和反向传播中的梯度弥散问题；同时相较于包含 3 个门信息的 LSTM 单元，GRU 具有更少的训练参数，在保证精度的情况下优化了训练中反向传播的计算效率。

在本文的编码器中，每个 GRU 的重置门有针对性地记忆当前视频帧图片特征的信息，更新门则调节上一视频帧有效信息的保留比例，最终输出当前视频帧的有效信息。图 3 展示了单个 GRU 的内部结构。图中 h^{t-1} 表示第 $t-1$ 帧的有效信息， x^t 表示第 t 帧的图片特征， y^t 和 h^t 表示第 t 帧的有效信息，r 和 u 分别表示重置门信息和更新门信息。

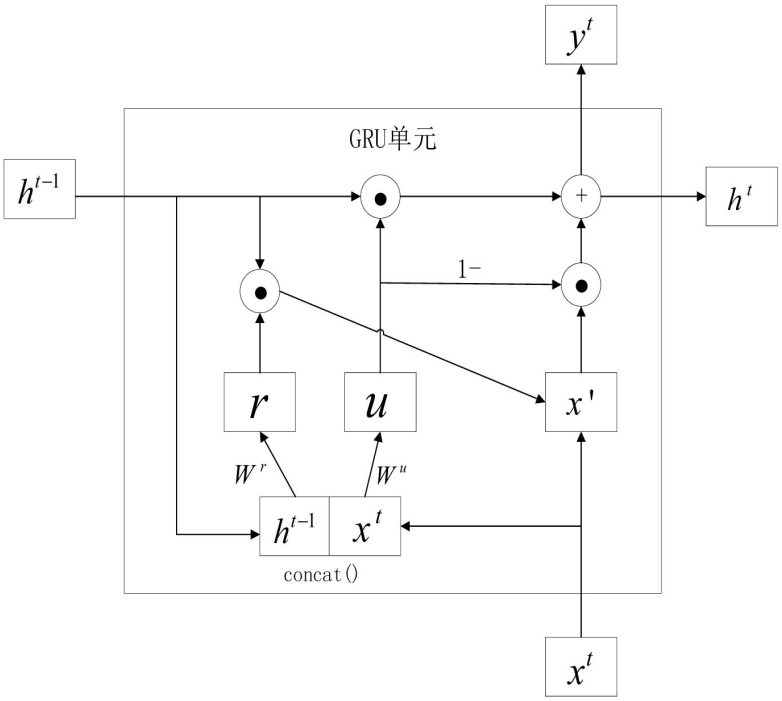

Figure 3. The structure of GRU (⊙ represents the dot product operation, ⊕ represents the addition operation)
图 3. GRU 结构(⊙ 表示点乘操作，⊕ 表示相加操作)

GRU 的工作过程分为三步：首先，根据第 t 帧的图片特征 x^t 和第 $t-1$ 帧的有效信息 h^{t-1} 来获取重置门信息 r 和更新门信息 u：

$$r=\sigma\left(W^r\cdot concat\left(x^t,h^{t-1}\right)\right) \tag{1}$$

$$u=\sigma\left(W^u\cdot concat\left(x^t,h^{t-1}\right)\right) \tag{2}$$

其中 $concat(\,)$ 表示将两个向量或矩阵进行拼接，W^r 和 W^u 是需要训练的权重参数，$\sigma(\,)$ 表示使用 sigmoid 函数归一化。然后，使用重置门 r 筛选出上一视频帧对当前帧的有用数据 x'：

$$x' = \tanh\left(W^x \cdot concat\left(x^t, h^{t-1} \cdot r\right)\right) \tag{3}$$

其中 $h^{t-1} \cdot r$ 表示将第 $t-1$ 帧的有效信息 h^{t-1} 和所得重置门信息 r 进行点乘，$\tanh(\,)$ 表示通过双曲正切函数将结果约束在(−1, 1)之间。最后，使用所得更新门信息 u 计算第 t 帧的有效信息 y^t：

$$h^t = u \cdot h^{t-1} + (1-u) \cdot x' \tag{4}$$

$$y^t = h^t \tag{5}$$

其中 $u \cdot h^{t-1}$ 进行遗忘操作，将第 $t-1$ 帧的信息选择性遗忘，$(1-u) \cdot x'$ 进行记忆操作，将第 t 帧的图片特征选择性记忆。最终得到的 h^t 用于计算第 $t+1$ 帧的有效信息，y^t 作为当前 GRU 的输出。

由公式(4)可以看出，每个 GRU 包含的其他帧的信息均以加算的形式保留在当前状态中，因此在反向传播时，每一个过去状态的相应影响权重不会趋向于 0，避免梯度弥散问题。

针对视频帧序列的特性，本文不仅要考虑当前帧之前的视频帧对当前帧的影响，也要考虑当前帧之后的视频帧对当前帧的影响，故采用双向 GRU 网络作为获取帧级重要性得分模块的编码器。双向 GRU 网络包括 forward 层和 backward 层。经过特征提取的视频帧序列输入到双向 GRU 网络后，forward 层从 F_0 到 F_{n-1} 正向计算并保存当前帧之前各帧对当前帧的影响信息，backward 层从 F_{n-1} 到 F_0 反向计算并保存当前帧之后各帧对当前帧的影响信息。最后在每个时刻结合 forward 层和 backward 层的相应时刻输出的结果得到的当前帧最终的有效信息。在编码器中，通过 GRU 计算第 t 帧的有效信息 o_t：

$$h_f^t = f_{GRU}\left(x^t, h^{t-1}\right) \tag{6}$$

$$y_f^t = h_f^t \tag{7}$$

$$h_b^t = f_{GRU}\left(x^t, h^{t+1}\right) \tag{8}$$

$$y_b^t = h_b^t \tag{9}$$

$$o_t = \sigma\left(W_{fo} y_f^t + W_{bo} y_b^t\right) \tag{10}$$

其中 $f_{GRU}(\,)$ 表示根据第 t 帧的图片特征和第 $t-1$ 帧的有效信息，经过使用公式(1)-(5)的过程计算获得第 t 帧的正向(或反向)有效信息。y_f^t 和 h_f^t 表示 forward 层中第 t 帧的正向有效信息，y_b^t 和 h_b^t 则表示 backward 层中第 t 帧的反向有效信息，通过公式(10)对两层的输出加权求和，并使用 sigmoid 函数对结果进行归一化，获得第 t 帧的最终有效信息 o_t。

3.2.2. 解码器改进：结合注意力机制获得重要性得分

为了解决有损压缩的中间向量难以存储足够信息的问题，本文在解码器部分结合注意力机制对视频帧信息进行解码。

注意力机制[21]通过借鉴人类视觉注意力的工作原理，在序列中筛选出重要性更高的部分作为当前节点输出的判断依据。在本文中，注意力机制模块首先通过快速扫描全局视频帧，获得需要重点关注的几个目标帧，然后对上述目标帧投入更多注意力资源，以获取更多对当前帧的重要性评判的信息，同时抑制非目标帧的无用信息。除了加入注意力机制之外，本文在解码器部分同样采用 GRU 网络，进一步减少参数的数量，提高计算效率。

本文的解码器解码过程如图 4 所示。根据在编码器中获得的每一帧的有效信息 o_i，结合解码器每个时刻的状态 S_j，可以求得第 j 时刻的注意力信息 $context_j$。

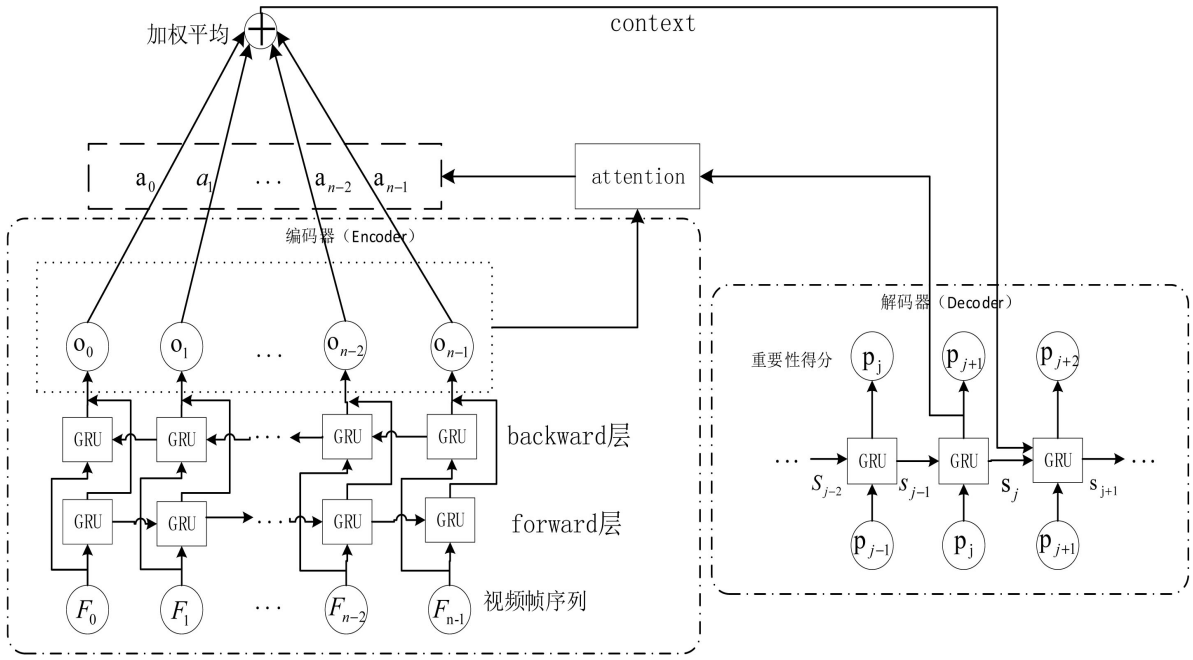

Figure 4. The process of obtaining the importance score of video frames combined with the GRU network and the attention mechanism

图 4. 结合 GRU 网络和注意力机制的获取视频帧重要性得分过程

$$\alpha_{i,j}=\frac{\exp\left(e\left(o_i,s_j\right)\right)}{\sum_i \exp\left(e\left(o_i,s_j\right)\right)} \tag{11}$$

$$context_j=\sum_i \alpha_{i,j}o_i \tag{12}$$

其中 $\alpha_{i,j}$ 表示其他每一帧 i 对第 j 帧的注意力权重，$e\left(o_i,s_j\right)$ 是相关度函数，衡量当前帧与解码器状态的相关度，本文采用的相关度函数是：

$$e(o,s)=X\tanh\left(Yh+Zs\right) \tag{13}$$

其中 X,Y,Z 是模型参数。$context_j$ 反映了对当前视频帧最相关的信息。进一步地，利用所得 $context_j$ 可以计算出解码器 $j+1$时刻的状态 s_{j+1} 以及 $j+1$时刻的重要性得分 p_{j+2}：

$$s_{j+1}=f_{GRU}\left(p_{j+1},concat\left(s_j,context_j\right)\right) \tag{14}$$

$$p_{j+2}=s_{j+1} \tag{15}$$

通过将 j 时刻的状态 s_j 和第 j 帧的注意力信息 $context_j$ 拼接后作为 $j+1$时刻的输入，解码器可以在解码的每一步查询最相关的原视频有效信息，避免中间向量信息不足问题。

3.3. 基于非极大值抑制生成视频摘要

现有方法通常使用 DPP 算法或动态规划的方法生成视频摘要，两者所选取的视频摘要均具有不错的概括能力，但是这些方法主要通过重要性得分的高低筛选关键帧，而没有关注所选取的关键帧之间的特征相似程度，因此在生成的视频摘要中仍存在一定程度的冗余。在视频摘要生成模块中，与现有模型不同，本文将去除冗余帧作为主要任务，基于非极大值抑制的思想设计算法方案，在筛选关键帧时，不仅

考虑重要性得分更高的视频帧，同时根据帧与帧之间的图片相似度，过滤与已选视频帧相似度高于阈值的视频帧，使在关键帧序列中重要性得分总和尽可能高的前提下，保证关键帧之间的相似度尽可能低。

在生成关键帧序列的步骤中，按照算法 1，遍历上一模块获得的帧级重要性得分序列，采用基于非极大值抑制的算法去除冗余帧，生成关键帧序列。

算法 1 的第 2 行将帧级重要性得分序列 P 重新排序，使遍历序列 P 时优先筛选得分更高的帧。在遍历序列 P 过程中，5~7 行限制关键帧序列 K 的长度，保证 K 的长度不大于 m；9~14 行计算当前帧 *curframe* 与关键帧序列 K 中每一帧之间的相似度并与阈值 α 比较，第 10 行中 $sim\left(Z\left[curframe\right],Z\left[kframe\right]\right)$ 表示根据 *curframe* 和 *kframe* 的特征向量采用余弦距离计算相似度；若上一步中与 K 中每一个关键帧的相似度均低于 α，则将 *curframe* 加入 K 中；20 行将 K 按时间顺序重新排列，作为所提模型的静态视频摘要。

进一步地，根据上一模块得到的关键帧序列 K 和帧级重要性得分 P，按照算法 2，筛选出关键帧对应的关键镜头，并根据当前关键镜头序列的总时长情况，决定删减或扩充镜头时长。与文献[14]和文献[15]相同，本文采用基于核的时域分割(kernel temporal segmentation，KTS) [9]镜头检测算法获取镜头序列 P'，并且将最后生成的关键镜头序列总时长 n 限定在原视频长度的 15%左右。

算法 2 的 5~10 行通过遍历关键帧序列 K，将所有关键帧对应的镜头加入关键镜头序列 S 中；12~23 行通过删减 K 中不含关键帧的镜头以解决 K 时长过长的情况；25~35 行通过贪心算法补充镜头以解决 K 时长不足的情况，其中第 26 行的镜头级重要性得分由镜头中所有帧得分相加获得。

Algorithm 1. Algorithm for getting key frame sequence
算法 1. 获取关键帧序列算法

输入：帧级重要性得分序列 P 和帧特征序列 Z
输出：关键帧序列 K

1：**begin**
2：按重要性得分降序对序列 P 重新排序；
3：初始化关键帧序列 K、K 的最大容量 m 和相似度阈值 α；
4：**for** (each *curframe* in P) **do**
5：　**if** K 当前长度 >=m **then**
6：　　**break;**
7：　**end if**
8：　tag ← 0；
9：　**for** (each *kframe* in K) **do**
10：　　**if** $sim\left(Z\left[curframe\right],Z\left[kframe\right]\right)>\alpha$ **then**
11：　　tag ← 1；
12：　　**break;**
13：　　**end if**
14：　**end for**
15：　**if** tag == 1 **then**
16：　　**break;**
17：　**end if**
18：　将 *curframe* 的 key 值添加到 K 中；
19：**end for**
20：按帧序号升序对 K 重新排序；
21：return K；
22：**end**

Algorithm 2. Algorithm for getting key shot sequence
算法 2. 获取关键镜头序列算法

输入：帧级重要性得分序列 *P* 和关键帧序列 *K*

输出：关键镜头序列 *S*

begin
使用 KTS 标记出原视频的镜头分割帧，得到镜头序列 *P'*；
按重要性得分降序对 *P'*序列重新排序；
初始化关键镜头序列 *S* 和 *S* 的最大时长 *n*；
for (each *kframe* in *K*) **do**
    根据序列 *P'*找到 *kframe* 的对应镜头 *kshot*；
    if *kshot* 不在 *S* 中 **then**
        将 *kshot* 加入 *S* 中；
    end if
end for
if *P* 的总时长 >= *n* **then**
    初始化超出的时长 *overtime*；
    while *overtime* > 0 **do**
        for(each *curp* in *P*) **do**
            根据帧序号找到当前关键帧的位置；
            记镜头 *curp* 含有 *x* 个关键帧，将镜头 *curp* 平均分成 *x*+1 个镜头，删除不含关键帧的镜头，将剩余镜头组合成新的镜头赋值给 *curp*，将所删除的时长记为 *deltime*；
            if *deltime* > *overtime* **then**
                break；
            else
                overtime ← *overtime* - *deltime*；
            end if
        end for
    end while
else
    初始化不足的时长 *lefttime*；
    通过 *P* 生成镜头级重要性得分序列 P_{shot}；
    按照重要性得分降序对 P_{shot} 重新排序；
    for (each curp in P_{shot}) **do**
        将 *curp* 加入 *S*；
        if *curp* 的时长 > *lefttime* **then**
            break；
        else
            lefttime ← *lefttime* - *curp* 的时长；
        end if
    end for
end if
按时间顺序对 *S* 重新排序；
return *S*；
end

4. 实验分析

4.1. 实验数据集

本文实验用到 3 个公开数据集，分别是 TVSum [22]、SumMe [23]、和 VSUMM [5]。

TVSum 数据集包含 50 个视频，包含车辆修理、派对聚会、宠物展示等类型的内容。SumMe 数据集包含 25 个视频，包含体育竞赛、节日庆祝等主题，且视频均是未做后期处理的拍摄视频。VSUMM 数据集包含 100 个经过后期处理的视频，时长均在 10 分钟以下，其中 50 个视频来自 YouTube 网站，包含卡通，新闻，体育，商业广告，电视节目和家庭视频这 6 种类型，另外 50 个视频来自 open video project 网站，主要以纪录片为主。TVSum 和 SumMe 的标签是由 20 名测试者人工标注的重要性评分，VSUMM 的标签是由 5 名测试者人工选取的关键帧。

为了更有效地训练模型，本文对这些数据集的标签进行转换。其中 TVSum 和 SumMe 的标签需要选择这 20 个人工标注的重要性得分的平均值，并缩放到区间[0, 5]之内；而 VSUMM 的标签则需要把相应的关键帧转化为重要性得分，最大值为 5(即 5 名测试者都选择了该帧为关键帧)，最小值为 0 (即 5 名测试者均未选择该帧)。

为了便于与其他模型作比较，本文采用大多数主流模型的数据集划分方法，即所有数据集均按照 6:2:2 的比例划分为训练集、验证集和测试集；同时将可能影响模型整体性能的卡通视频去除，即将 VSUMM 中的 11 个卡通视频移除数据集，保留剩余的 89 个视频。

4.2. 模型评价指标

为本文采用的评价指标包括 *F-score*、关键帧冗余率 *KFRR*。

F-score 用于判断模型所生成视频摘要的整体表征能力。*F-score* 由精确率 *pre* 和召回率 *rec* 计算得出：

$$pre = \frac{S_{sim}}{S_m} \tag{16}$$

$$rec = \frac{S_{sim}}{S_t} \tag{17}$$

$$F\text{-}score = \frac{2 * pre * rec}{pre + rec} \tag{18}$$

其中 S_{sim} 表示模型所生成的视频摘要与测试数据的视频摘要的重叠率，即生成摘要与测试摘要匹配的长度，S_m 表示模型所生成的视频摘要的总长度，S_t 表示测试数据的视频摘要的总长度。精确率衡量模型所生成摘要的总长度。精确率衡量模型所生成摘要相对于测试数据摘要的准确性，召回率衡量模型所生成摘要在测试数据摘要中的比重，是精确率和召回率的调和平均数，衡量两者的整体水平，即衡量模型所生成视频摘要的整体质量。

为了评估所提模型的去冗余效果，本文提出一种新的评估指标——关键帧冗余率，衡量在生成的关键帧序列中，图片的整体重叠率。本文将所生成关键帧序列的整体图片相似度，作为模型的整体关键帧冗余率：

$$sim(x, y) = \tanh\left(\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}\right) \tag{19}$$

$$R_{cur} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{m} sim(cur_i, cur_j)}{(m-1)!} \tag{20}$$

$$KFRR = \frac{\sum_{i=1}^{z}(R_i * T_i)}{T_{sum}} \tag{21}$$

公式(19)中 $sim(x, y)$ 表示关键帧 x 与 y 的相似度，x_i 和 y_i 表示长度为 n 的图片特征向量中的每个值。公式(20)中 R_{cur} 表示模型对当前测试视频的关键帧冗余率，通过计算每个关键帧之间的相似度的平均值获得。公式(21)中 R_i 表示当前测试视频的冗余率(即 R_{cur} 的计算结果)，T_i 表示当前测试视频的时长，T_{sum} 表示所有测试视频的总时长，通过将每一个测试视频的冗余率加权平均获得 *KFRR*，其值越小，表示模型的去冗余效果越好。

4.3. 实验过程和结果分析

本文实验采用 opencv 工具提取视频帧，并将视频帧序列下采样(每秒保留 2 帧)。特征提取模块采用预训练后的 googLeNet 网络；重要性得分获取模块由编码器和解码器构成，编码器由 2 个不同传输方向的 GRU 网络层组成，每层包含 256 个 GRU，解码器由 1 个 GRU 网络层和注意力获取模块组成；视频摘要生成模块则采用 3.3 节所提算法生成摘要，根据上一模块获取到的重要性得分最大长度设置为 8 和相似度阈值设置为 0.6。训练过程中，损失函数采用均方差函数，优化算法选择随机梯度下降算法，BatchSize 设置为 32，学习率设置为 0.005。本文所提模型的视频摘要的生成流程如图 5 所示。

Figure 5. Process diagram of summarizing video
图 5. 本文模型的视频摘要生成过程图

实验中用到的对比模型是 VSUMM [5]、dppLSTM [14]和 SUM-attDecode [15]。其中 VSUMM 采用 k-均值聚类算法生成静态视频摘要，dppLSTM 和 SUM-attDecode 采用基于 LSTM 的 Seq2Seq 模型生成静态摘要和动态摘要，其中 dppLSTM 使用 DPP 算法去除冗余帧，SUM-attDecoder 则使用动态规划算法去除冗余帧。

4.3.1. 比较 *F-score*

本文首先模拟小数据集下模型的性能，仅使用上述 3 种数据集中的一种用于训练和测试模型，最终结果如表 1 所示。

Table 1. Comparison of *F-score* of different models under small dataset
表 1. 小数据集下不同模型的 *F-score* 对比结果

数据集	模型	*F-score*
TVSum	dppLSTM	54.7
	SUM-attDecoder	52.9
	本文模型	54.9
SumMe	VSUMM	33.7
	dppLSTM	38.6
	SUM-attDecoder	38.2
	本文模型	39.2

表 1 展示了在 SumMe 和 TVSum 两种数据集中，本文模型与各种主流视频摘要模型的 *F-score* 值对比结果。从表 1 可以看出，本文模型在 SumMe 和 TVSum 中都具有较高的 *F-score* 值。相比于 dppLSTM 和 SUM-attDecoder，本文模型所采用的 GRU 参数更少，因此在数据集较少的情况下，可以采用更高的学习率，使模型快速收敛，从而达到更好的效果。

进一步地，本文通过数据增强，模拟大数据集下模型的性能，与文献[14]和文献[15]相似，将分别随机选取 SumMe 和 TVSum 中 20%数据作为测试集，将剩余的 80%数据和另外 2 种数据集合并作为训练集和验证集，数据增强后的实验结果如表 2 所示。

Table 2. Comparison of *F-score* of different models under large dataset
表 2. 大数据集下不同模型的 *F-score* 对比结果

训练数据集和验证数据集	测试数据集	模型	*F-score*
SumMe + VSUMM + 80%TVSum	20% TVSum	dppLSTM	59.6
		SUM-attDecoder	58.9
		本文模型	59.7
TVSum + VSUMM + 80%SumMe	20% SumMe	dppLSTM	42.9
		SUM-attDecoder	44.0
		本文模型	44.3

从表 2 可以看出，相比于小数据集，本文模型在大数据集中的 F-score 提高了 5 到 6 点，证明本文模型在有充足的训练数据和验证数据的情况下，能收敛到更优的结果。同时对比其他模型在数据增强后的结果，再次证明本文模型具有更高的性能。

4.3.2. 比较 KFRR

由于 KFRR 评估指标需要根据模型获取到的关键帧进行计算，因此实验中选择 VSUMM 和 dppLSTM 作为比对模型，对比结果如表 3 所示。

Table 3. Comparison of *KFRR*
表 3. *KFRR* 对比结果

数据集	模型	*KFRR*
TVSum	VSUMM	0.71
	dppLSTM	0.59
	本文模型	0.53
SumMe	VSUMM	0.78
	dppLSTM	0.56
	本文模型	0.45

由表 3 可以看出，本文模型的 KFRR 值在两个数据集中都保持较低水平，说明本文模型具有更好的去冗余效果。由于 SumMe 数据集均是原生视频，含有大量的冗余帧，因此本文模型在 SumMe 数据集中的去冗余效果更加明显。

5. 结语

本文提出了一种融合 GRU 和非极大值抑制的视频摘要生成模型。所提模型在原有 Seq2Seq 模型基础上，使用 GRU 替代传统循环神经网络单元，有效地减少训练参数数量，并使用非极大值抑制算法更高效地去除冗余帧。实验数据表明，所提模型具有较优的性能和去冗余效果，同时具有更好的拟合效果，即使在小数据集中也具有不错的性能。

目前所用数据集均为短视频，导致所训练模型对长视频的摘要生成效果不理想，未来的研究可以考虑通过分段检测的方式对长视频进行逐段检测，并在此基础上对分段视频之间的关联关系进行建模，使视频摘要生成模型能够适应长镜头的检测工作。

致　谢

以上是论文的全部论述内容，在这里感谢广东省科技计划项目给予资金支持，感谢陈平华老师的实验指导和修改意见，同时论文的成功撰写离不开实验室小伙伴的帮助，忠心感谢师生给予的良好学术环境。

基金项目

广东省科技计划项目(No.2020B1010010010、No.2019B101001021)。

参考文献

[1] 刘波. 视频摘要研究综述[J]. 南京信息工程大学, 2020, 12(3): 274-278.

[2] Amiri, A. and Fathy, M. (2010) Hierarchical Keyframe-Based Video Summarization Using QR-Decomposition and Modified-Means Clustering. *EURASIP Journal on Advances in Signal Processing*, **2010**, Article ID: 892124. https://doi.org/10.1155/2010/892124

[3] Guimaraes, S.J.F. and Gomes, W.A. (2010) Static Video Summarization Method Based on Hierarchical Clustering. In: *Ibero-American Congress Conference on Progress in Pattern Recognition*, Springer-Verlag, Berlin, 46-54. https://doi.org/10.1007/978-3-642-16687-7_11

[4] Frey, B.J. and Dueck, D. (2007) Clustering by Passing Messages between Data Points. *Science*, **315**, 972-976. https://doi.org/10.1126/science.1136800

[5] de Avila, S.E.F. and Lopes, A.P.B. (2011) VSUMM: A Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern Recognition Letters*, **32**, 56-68. https://doi.org/10.1016/j.patrec.2010.08.004

[6] Mundur, P., Rao, Y. and Yesha, Y. (2006) Keyframe-Based Video Summarization Using Delaunay Clustering. *International Journal on Digital Libraries*, **6**, 219-232. https://doi.org/10.1007/s00799-005-0129-9

[7] Khosla, A., Hamid, R., Lin, C.J., *et al*. (2013) Large-Scale Video Summarization Using Web-Image Priors. 2013 *IEEE Conference on Computer Vision and Pattern Recognition*, Portland, 23-28 June 2013, 2698-2705. https://doi.org/10.1109/CVPR.2013.348

[8] Panda, R. (2017) Weakly Supervised Summarization of Web Videos. 2017 *IEEE International Conference on Computer Vision*, Venice, 22-29 October 2017, 3677-3686. https://doi.org/10.1109/ICCV.2017.395

[9] Potapov, D., Douze, M., Harchaoui, Z., *et al*. (2014) Category-Specific Video Summarization. *European Conference on Computer Vision*, Zurich, 6-12 September 2014, 540-555. https://doi.org/10.1007/978-3-319-10599-4_35

[10] Zhang, K., Chao, W.L., Sha, F., *et al*. (2016) Summary Transfer: Exemplar-Based Subset Selection for Video Summarization. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 1059-1067. https://doi.org/10.1109/CVPR.2016.120

[11] Gygli, M., Song, Y. and Cao, L. (2016) Video2GIF: Automatic Generation of Animated Gifs from Video. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 1001-1009.

https://doi.org/10.1109/CVPR.2016.114

[12] Sun, M., Farhadi, A. and Seitz, S. (2014) Ranking Domain-Specific Highlights by Analyzing Edited Videos. In: *European Conference on Computer Vision*, Springer, Berlin, 787-802. https://doi.org/10.1007/978-3-319-10590-1_51

[13] Zhao, B., Li, X.L. and Lu, X.Q. (2017) Hierarchical Recurrent Neural Network for Video Summarization. In: *The* 2017 *ACM on Multimedia Conference*, ACM, New York, 863-871. https://doi.org/10.1145/3123266.3123328

[14] Zhang, K., Chao, W.L., Sha, F., *et al*. (2016) Video Summarization with Long Short-Term Memory. In: *European Conference on Computer Vision*, Springer, Berlin, 766-782. https://doi.org/10.1007/978-3-319-46478-7_47

[15] 冀中, 江俊杰. 基于解码器注意力机制的视频摘要[J]. 天津大学学报(自然科学与工程技术版), 2018, 51(10): 31-38.

[16] Mahasseni, B., Lam, M. and Todorovic, S. (2017) Unsupervised Video Summarization with Adversarial LSTM Networks. *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 2982-2991. https://doi.org/10.1109/CVPR.2017.318

[17] Yang, H., Wang, B.Y., Lin, S., *et al*. (2015) Unsupervised Extraction of Video Highlights via Robust Recurrent Auto-Encoders. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 4633-4641. https://doi.org/10.1109/ICCV.2015.526

[18] Sutskever, I., Vinyals, O. and Le, Q.V. (2014) Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, **32**, 3452-3462.

[19] Szegedy, C., Liu, W., Jia, Y., *et al*. (2014) Going Deeper with Convolutions. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 1-9. https://doi.org/10.1109/CVPR.2015.7298594 https://ieeexplore.ieee.org/document/7298594

[20] Cho, K., Merrieenboer, B., Gulcehre, C., *et al*. (2014) Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *Conference on Empirical Methods in Natural Language Processing* (*EMNLP* 2014), Doha, 25-29 October 2014, 1724-1734. https://doi.org/10.3115/v1/D14-1179

[21] Bahdanau, D., Cho, K. and Bengio, Y. (2015) Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representation*, San Diego, 7-9 May 2015, 1334-1349.

[22] Song, Y., Vallmitjana, J., Stent, A., *et al*. (2015) TVSum: Summarizing Web Videos Using Titles. 2015 *IEEE Conference on Computer Vision and Pattern Recognition*, Boston, 7-12 June 2015, 5179-5187.

[23] Gygli, M., Grabner, H., Riemenschneider, H., *et al*. (2014) Creating Summaries from User Videos. In: *European Conference on Computer Vision*, Springer, Cham, 505-520. https://doi.org/10.1007/978-3-319-10584-0_33